**Data Integration and Data Flow Modeling with SAP Data Warehouse Cloud**

ANA365

Exercises  
Jascha Kanngiesser / SAP SE

Table of ContentS

[Systems 3](#_Toc56606232)

[Participating live 3](#_Toc56606233)

[On Demand 3](#_Toc56606234)

[Things you will learn in this session 3](#_Toc56606235)

Thank you for participating in this hands-on session about Data Integration and Data Flow Modeling with SAP Data Warehouse Cloud! The session is divided into three main parts:

1. Space creation and connection setup
2. Data loading and modeling
3. Data visualization

# Systems

For working your way through the hands-on materials you need access to a SAP Data Warehouse Cloud tenant. Depending on whether you join the live session or watch the session on demand or replay the materials at a later point in time, different systems and credentials can be used.

Participating live

In case you are reading this manual while participating in the live session during the SAP TechEd 2020 program, you can use any of the following three tenants for participating in this hands-on session:

* EMEA region: <https://dwc-teched2020.eu10.hcs.cloud.sap/>
* APJ region: <https://dwc-teched2020.ap10.hcs.cloud.sap/>
* NA region: <https://dwc-teched2020.us10.hcs.cloud.sap/>

To receive login credentials please send an e-mail to [jascha.kanngiesser@sap.com](mailto:jascha.kanngiesser@sap.com?subject=Please%20share%20login%20credentials%20for%20session%20ANA365%20-%20Data%20Flow) mentioning the session ID and the tenant you would like to log in to.

On Demand

In case you are working on this hands-on material after the live session you cannot use any of the three tenants listed above. Instead, you need to get yourself a SAP Data Warehouse Cloud trial tenant here: <https://saphanajourney.com/data-warehouse-cloud/trial/>

You need to set up all the required connectivity yourself, including the Data Provisioning Agent, uploading certificates, creating the data sets in the remote sources (or uploading the data sets to SAP Data Warehouse Cloud).

Please note that the exercise below is written for attending the live session. Therefore, for example the user credentials and tenant information might not match in case you are following the materials later on demand. In this case please substitute any live session-specific information with the individual information available to your in your trial tenant.

# Things you will learn in this session

* How to create federated and replicated connections to SAP & non-SAP, cloud and on-premises data sources.
* How to load data into SAP Data Warehouse Cloud using external SQL clients.
* How to upload CSV files.
* How to model ETL processes to extract data from external sources into SAP Data Warehouse Cloud.
* How to consume & visualize data models using the SAP Analytics Cloud story builder and external SQL clients.

| Explanation | Screenshot |
| --- | --- |
| 1. Make sure you have the login credentials available to you. You should have received a username like saptechedana365+**xyz**@sapcom and a password. Replace the value **xyz** with the number assigned to you. |  |
| 1. Let’s get going – first part! | As mentioned above, this session consists of three main areas: Creating your space and required connections, load the data and build your data models and visualizing the results. Let’s get started with creating your space and setting up the required connectivity. |
| 1. Click on the menu bar to expand the navigation menu. |  |
|
| 1. Select Space Management in the bottom left corner. |  |
| 1. Hit the Create Space + - button in the top right corner to create a new space. |  |
| 1. Enter a Space Name and Space ID. Make sure that at least the Space ID follows the syntax ANA365\_<your three-digit number>. Then hit Create to create your space. |  |
| 1. Reduce the space size in the Overview section at the top. Make sure to specify both the Disk (GB) and In-Memory (GB) storage assignment as 0,1 (GB). |  |
| 1. Hit Save in the top right corner to sabe your changes. |  |
| 1. Head on to the Members section and hit the Add button on the right to add yourself to your space. |  |
| 1. Search for your user ANA365\_<your three digit number> in the dialog, select your user and hit Add to close the dialog. **Make sure that it is really your ANA365 user!** |  |
| 1. Head on to the Connections (Local Connections) section and hit the + - button to create a new connection. |  |
| 1. Select the Google Cloud Storage tile. |  |
| 1. Or use the filter and search for Google Cloud storage in case you cannot find it in the list of connections. |  |
| 1. Enter a meaningful business name and technical name as well as a useful description. |  |
| 1. Hit Next Step to navigate to the next screen. |  |
| 1. Enter the Project as flash-ocean-262507 |  |
| 1. Enter the root path as /digital-workshop-series/ |  |
| 1. Download the key gcs-key.json from the assets folder in the Github repository here: <https://github.com/SAP-samples/teched2020-ANA365/tree/main/assets/step-gcs-key> |  |
| 1. Hit the Browse button next to the Key entry field and select the downloaded file. |  |
| 1. Hit Create Connection to finish the dialog. |  |
| 1. Select the created connection from the list of connections and hit the Validate Connection button on the right. |  |
| 1. Make sure that the connection is valid for data flow building in the toast message show non the bottom of the screen. |  |
| 1. Hit the + button again to create another connection. |  |
| 1. This time select the SAP BW connection. |  |
| 1. In case you cannot find it, search for SAP BW in the filter bar. |  |
| 1. Enter a meaningful business name and technical name as well as a descriptive description. |  |
| 1. Make sure that the right Data Provisioning Agent is selected. The ending (in this case us10) should match the tenant you are logged in to. |  |
| 1. Hit Next Step. |  |
| 1. Fill out the Connection Details. Use ld2529 as the Application Server. |  |
| 1. Enter Client 100. |  |
| 1. Optionally enter the language as EN (is the default anyway). |  |
| 1. Enter Systen Number 20. |  |
| 1. Leave the Streamind Read option as Off. Enter user CLOUDCON with password Teched2020 |  |
| 1. Hit Create Connection to finish the dialog. |  |
| 1. Select the newly created SAP BW connection and again hit the Validate Connection button on the right. |  |
| 1. Make sure that the toast message on the bottom of the screen mentions that the connection can be used in the view builder. |  |
| 1. Head on to the Database Access (Database Users) section. Hit the Create button to create a new database user. |  |
| 1. Enter a meaningful Database User Name suffix and make sure to select the Enable Data Ingestion checkbox in the Privileges section. Then hit Create to close the dialog. |  |
| 1. Make sure to copy the Host Name, Port, Database User Name and Password to a safe place, for example your local notes on Mac or Notepad on Windows. Then hit Close to finish the user creation. |  |
| 1. The created database user should be shown with status **Active**. |  |
| 1. Congratulations! | You successfully created your space and created connections to a non-SAP source, Google Big Query! Also you integrated a SAP BW system and enabled your space to get connected to external SQL clients! Before we continue with the second parts, let’s quicky make sure that the space is still empty. |
| 1. Before we start loading data into our space, hit the Monitor Space button in the top right corner. |  |
| 1. Make sure that the space does not yet contain any data. |  |
| 1. Hit Edit Space in the top right corner to navigate back to the Management of your Space. |  |
| 1. Now it’s time to prep your space with some data! | As part of the second section of this hands-on we will equip your space with all the required data and setup the data models. We will ingest some data via an external SQL client (SAP HANA Database Explorer, but can be any client actually), replicate data from some of the data sets available in the SAP BW source, upload a flat file, use the Data Flow to ETL-like move data from Google Cloud Storage into your space and combine the locally available data from these different steps with some data we access virtually from the SAP BW system. |
| 1. Navigate back to the Database Access section and select the Database User you created. |  |
| 1. Hit the Open Database Explorer button on the right. This takes you to the SAP HANA Database Explorer website. |  |
| 1. If you have enabled two-factor authentication on SAP Cloud Platform you need to enter your passcode first. If not enabled you will be directly forwarded to the screen shown in the next step. |  |
| 1. Within the SAP HANA Database Explorer you can execute DML and DDL statements to create tables or views or insert table into tables. |  |
| 1. Hit the + button in the top left corner. |  |
| 1. From the Database Type select SAP HANA Database. |  |
| 1. Enter the information copied earlier to a safe place when creating the database user like the Host Name, Port, Database User and Password. |  |
| 1. Make sure to check the Save password (stored in the SAP HANA secure store) and Connect to the database securely using TLS/SSL (prevents data eavesdropping) options. Uncheck the Verify the server’s certificate using the trusted certificate below option. |  |
| 1. Hit the OK button to add the database to your list of databases. |  |
| 1. Right-click the newly created database and select Open SQL Console. |  |
| 1. Get the SQL statements from the folder in the Github repository to create and fill the table that holds the Sales Orders: <https://github.com/SAP-samples/teched2020-ANA365/tree/main/assets/step-sql-statements> |  |
| 1. Open the create-sales-orders.sql file and copy over the content in the SQL console opened in the SAP HANA Database Explorer. |  |
| 1. Hit the green Execute button to create the Sales Order table. |  |
| 1. Make sure that the table was created correctly by checking the log. It should similar to what you see below, stating that the statement was executed (no errors mentioned). |  |
| 1. Empty the SQL console by removing the executed statements. |  |
| 1. Open the insert-sales-orders.sql file and copy over the SQL to fill the table you just created. |  |
| 1. Again hit the green Execute button and check the log for the successful execution. |  |
| 1. Congratulations! | You just used an external tool to connect to your SAP Data Warehouse Cloud space via SQL, created a table and inserted data! This is one example how you can use actually any external 3rd party tool, for example also open source applications like DBeaver, to connect to your SAP Data Warehouse Cloud space from the outside and ingest data using SQL. This way you can for example integrate your SAP Data Warehouse Cloud tenant in your already existing ETL processes. |
| 1. Get the required CSV file SalesOrderItems.csv from the Github repository: <https://github.com/SAP-samples/teched2020-ANA365/tree/main/assets/step-upload-csv-file> |  |
| 1. Head back to your SAP Data Warehouse Cloud tenant and navigate to the Data Builder. |  |
| 1. Select your space. |  |
| 1. On the right click the Import icon and select the Import CSV File option. |  |
| 1. Hit the Select Source File button and select the SalesOrderItems.csv file you just downloaded. |  |
| 1. Leave all options as is and hit the Upload button. |  |
| 1. Wait until the file was uploaded to the tenant. |  |
| 1. Make sure that the data is shown and the headers are identified correctly. |  |
| 1. Select the columns SALESORDERID and SALESORDERITEM as key fields. |  |
| 1. Make sure that both columns are enabled as key fields. |  |
| 1. Hit Deploy to open the dialog for deploying the table into your space. |  |
| 1. Enter a Business Name and Technical Name and deploy the table into your space. |  |
| 1. Make sure that the table was correctly created in your space and is visible in the Data Builder overview. |  |
| 1. Congratulations! | You have successfully uploaded data into your SAP Data Warehouse Cloud space using the CSV file upload! As a next step we will continue loading data into your space using the data flow feature. |
| 1. Select the New Data Flow tile. |  |
| 1. Switch to the Sources tab and navigate to the product reviews folder: Connections > Google Cloud Storage > digital-workshop-series |  |
| 1. Select the product-reviews.csv file and drag it from the tree on the left onto the canvas. |  |
| 1. Make sure that the CSV properties are set correctly by selecting the added node and expanding the CSV properties section on the right. |  |
| 1. In the Operators button bar above the canvas you can use several different operators for working with the data sets in your data flow. |  |
| 1. First, we want to calculate a sentiment from the available product reviews. Add the Script operator as a target node after the combined reviews source node. Please note that the left node should be named product-reviews in your case (this applies to the following screenshots, too). |  |
| 1. Connect the two nodes by dragging a line from the left to the right node. |  |
| 1. Select the Script node and navigate to the Script details by selecting the Edit button on the right. |  |
| 1. Select the Columns section. |  |
| 1. Add a new column named SENTIMENT with data type INT8. |  |
| 1. Confirm the details by clicking Save. |  |
| 1. Switch to the Script section and start entering your python script for a sentiment analysis. |  |
| 1. It’s up to you! | Here’s the thing: Are you a Python expert? We’ve got a challenge for you!  You can either try to figure out the right script yourself using the pandas library. Pro-tip: Check out the pandas data frame apply method documentation here: <https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.apply.html>  To make your life easier, we have already provided you with a list of predefined key words and assigned sentiments in the file assets/step-data-flow-script/sentiment-key-words.py  Using the list of words and assigned sentiment values (1: positive; -1: negative; word not mentioned: neutral / 0) you want to do two things:   1. Define a function that takes a row from the input data and checks whether the column REVIEW\_TITLE contains any of the words from the words sentiment array. If a match is found, you want to return the sentiment associated with this word, otherwise (no match) you want to return zero using the string.lower() and string.find() methods. 2. You want to use the pandas dataframe apply() function to loop over the data and call the function defined in 1) for each row. Make sure to specify the axis as 1 and the result\_type=reduce! |
| 1. Want to take the quick route? | Ok, fair enough – here’s the deal. You can use the pre-build solution from the assets/step-data-flow-script/sentiment-analysis-script.py file. If you do so, don’t put the blame on us that the solution is not the nicest or you finished the hands-on well before the scheduled 2hrs! ;-) |
| 1. Add a Projection node to filter out unwanted columns or to apply a filter on the data to be extracted. Make sure to connect it to the previous Script 1 node. |  |
| 1. Select the added Projection node and select Remove Column to hide the following columns from the Details screen on the right:  * REVIEW\_ID * CUSTOMER\_ID * COUNTRY * REVIEW\_DATE * REVIEW\_TITLE * REVIEW\_TEXT |  |
| 1. After removing all columns, only the following two columns should be available. |  |
| 1. Next, add an Aggregation node and connect it to the previous Projection 1 node. |  |
| 1. Select the node and from the Details pane on the right set the aggregation for the column SENTIMENT. |  |
| 1. Select the Add Table operator or use the context menu from the added aggregation node to add the target table in which we will persist the data. |  |
| 1. Select the newly added node and define a label, business name and technical name for the target table. |  |
| 1. Select the Create and Deploy Table button to create the table in the repository. |  |
| 1. Confirm the dialog to create the table. |  |
| 1. Change the mode from APPEND to TRUNCATE to make sure that in case of repetitive loads only the most recent data is available in the target table. |  |
| 1. This is how your data flow should look like by now. |  |
| 1. Save the data flow. |  |
| 1. Provide a name for the data flow and hit Save. |  |
| 1. After saving the data flow execute it. |  |
| 1. Navigate to the data flow monitor by selecting the navigation icon from the left to navigate to the run details. |  |
| 1. Hit the Refresh button until the data flow run completes. If the message Authorize us to run… is visible you can simply ignore it. ☺ |  |
| 1. If you wanted to, you could also define a schedule to run the data flow repeatedly. This is however not needed for this session. ☺ You can define schedules by selecting the Schedule item from the left next to the Refresh icon. |  |
| 1. You can specify different options like recurrence, start and end date. |  |
| 1. Make sure that your executed data flow executed completely. |  |
| 1. Navigate to the Data Builder to check whether the table created during the data flow execution is filled correctly. |  |
| 1. Select the Tables tab and select the Products Reviews table. |  |
| 1. Open the Data Preview by clicking on the Data Preview button in the View section at the top. Make sure the different products are listed and the sentiment is shown. The aggregated sentiment or product IDs might look a little different in your case from what is visible in this screenshot. |  |
| 1. Congratulations! | You have successfully created and executed your first Data Flow instance in SAP Data Warehouse Cloud! You now know how to create ETL processes, schedule data loads using data flows and how to transform the incoming data.  As a next step, let’s take a look at how you can integrate data virtually from the connected SAP BW system to add the required product master data to our data model. We will now build a virtual master data dimension. |
| 1. Navigate to the Views area in the Data Builder. |  |
| 1. Select the New Graphical View tile and select the Sources tab. |  |
| 1. Drill down into the SAP BW connection and navigate to SAP BW > Extractors > BW > P… |  |
| 1. Start off by dragging over the PCATID – Product Category – Master Data Texts. |  |
| 1. Provide a meaningful Business Name and Technical Name. |  |
| 1. Add a calculation node. |  |
| 1. Add a new calculated column. |  |
| 1. Provide a meaningful Business Name and Technical Name and define the Data Type as String. |  |
| 1. Select the column TXTSH from the list of columns as the Expression. |  |
| 1. Hit the Validate button to make sure the expression is valid. |  |
| 1. Add a projection node after the calculation node. |  |
| 1. Make sure to exclude the column Short Description and Language key. |  |
| 1. Next, drag over the PRODUCT – Product – Master Data Attributes from the Sources tree and put it onto the projection node. Make sure that the option Join is selected when dropping the table! |  |
| 1. Again, provide a meaningful Business Name and Technical Name. Then, click on Import and Deploy. |  |
| 1. Make sure that the Join mapping is defined correctly. |  |
| 1. Add a projection node after the Product Attributes. |  |
| 1. Exclude the columns Currency and Price. We don’t need these columns because the Sales Order data sets already contain this information. |  |
| 1. Next, drag over the PRODUCT – Product – Master Data Texts and drop it on the projection node you created after the Product Attributes table. |  |
| 1. Provide a meaningful Business Name and Technical Name. |  |
| 1. Add a projection node after the Product Texts node. |  |
| 1. Exclude the Language key column. |  |
| 1. Change the Business Name of the Short description column. |  |
| 1. Provide a meaningful name like Product Description. |  |
| 1. This is how your model should look like by now. |  |
| 1. Select the View 1 output node and open the data preview. Make sure that the data is displayed. |  |
| 1. In the Properties pane change the Type from Relational Dataset to Dimension. |  |
| 1. In the Attributes section select the hierarchy icon. |  |
| 1. In the dialog add a new level-based hierarchy. |  |
| 1. Add two levels and specify the first level as Product Category and the second level as Product. |  |
| 1. Confirm the dialog to create the hierarchy. |  |
| 1. Select the edit icon in the Attributes section. |  |
| 1. In the dialog change the Semantic Type for the columns Product Category Description and Product Description to Text. |  |
| 1. Change the Label Column values for the first row to Product Category Description and for the second row to Product Description. |  |
| 1. Confirm the dialog. |  |
| 1. Change the Business Name and Technical Name of the view to Products. |  |
| 1. Save and deploy the view. |  |
| 1. Congratulations! | You created your first master data dimension in SAP Data Warehouse Cloud! By now the dimension is all virtual. The data we put together in the dimension is fetched live from the connected SAP BW system whenever we query the data model, be it the data preview or when used later in a SAP Analytics Cloud story.  Next, we create our final data model combining all the different data assets in a single model. |
| 1. Navigate to the Data Builder and create a new Graphical View. |  |
| 1. We start with the Sales Order and Sales Order Items we created in the Open SQL Schema using the Database User. Switch to the Sources tab and drill down into the Open SQL Schema connection. Select the SalesOrders table and drag it onto the canvas. |  |
| 1. Provide a meaningful Business Name and Technical Name. |  |
| 1. Select the Repository tab and select the SalesOderItems table. Drag it onto the Sales Orders table and make sure the Join option is selected. |  |
| 1. Select the Join node and make sure the field mapping is correctly defined (Sales order ID -> SALESORDERID). |  |
| 1. Switch to the Repository tab and in the Tables list select the Product Reviews table. Drag it onto the canvas and drop it onto the Sales Order Items node. |  |
| 1. Select the join node and make sure that the join field mapping is correct (PRODUCTID -> PRODUCT\_ID) Don’t worry in case your output node on the right is still named View 1, we will get to that. |  |
| 1. In the General section make sure to set the Join Type as Left so that products for which no reviews exist are still available in the final result set. |  |
| 1. Do data preview on join node to validate that data is available. Again, the output might look a little different – don’t worry. ☺ |  |
| 1. Select the View 1 output node and scroll down to the Associations section. Hit the Add button to add a new association. The number of columns might be different from what you see in your tenant, but don’t worry. |  |
| 1. From the list of dimensions select the Products view and hit OK. |  |
| 1. In the details on the added association make sure that the join fields are defined correctly (PRODUCTID -> Product). |  |
| 1. Navigate back to the main Properties screen of the View 1 output node by clicking on the View 1 bread crump navigation at the top. |  |
| 1. Provide a meaningful Business Name and Technical Name and set the type to Analytical Dataset. Make sure to switch on the Exposing option. |  |
| 1. Now that you have set the type as Analytical Dataset you can define measures. Scroll down and move the columns   Sentiment, Gross amount, Net amount and Tax amount from the list of Attributes to the list of Measures. |  |
| 1. Select the edit button for the Attributes. Don’t worry if your view contains more than 21 attributes. |  |
| 1. Change the Semantic Type for the column Currency key to Currency Code. |  |
| 1. Select the edit button for Measures. |  |
| 1. Specify a readable business name for each of the measures. |  |
| 1. Change the Semantic Type for the three measures Gross amount, Net amount and Tax amount to Amount with Currency. Specify no semantic type for Sentiment. |  |
| 1. For the Unit Column select Currency key for the three measures expect Sentiment. |  |
| 1. Close the dialog. |  |
| 1. Save and deploy the view and confirm the validation message in case it is shown. |  |
| 1. Congratulations! | You created your data model and combined the different data assets. We are now ready to consume the data model in an application like SAP Analytics Cloud or a 3rd party external SQL application.  In the next section we will experience both possible ways, before we finally take a look at how the models can be tweaked further to improve the performance and make sure that the source systems are not overloaded with data requests.  We will start with creating a SAP Analytics Cloud story as an example, then take a look at how you can consume the exposed data models in a 3rd party SQL tool. |
| 1. Select the Story Builder from the navigation bar and create a new story. |  |
| 1. Select the Sales Data set you created earlier. If no entry is visible navigate back to the graphical Sales Data view and make sure the Exposing option is switched on! |  |
| 1. Select the Chart option. |  |
| 1. On the right add the two measures Gross amount and Sentiment. |  |
| 1. The aggregated sum and sentiment is shown on the left. |  |
| 1. Add the PRODUCTID dimension to the list of Dimensions. |  |
| 1. Change the PRODUCTID hierarchy display to Level 2. |  |
| 1. Drag the canvas a little wider to get a good overview on the aggregated sentiment per product category. |  |
| 1. Congratulations! | You have successfully consumed the data set in the SAP Analytics Cloud story builder. You can go ahead and tweak and fine-tune the story as you like, however we will not drill down deeper into the possibilities of the SAP Analytics Cloud story builder for the sake of this data integration session.  As a next step, let’s check out how you can consume the very same data model using a 3rd party SQL external application. |
| 1. Navigate back to the Space Management into your space to the Database Access section. Hit Create to create another Database User. |  |
| 1. Provide a Database User Name Suffix like DATA\_CONSUMPTION and make sure to select the Enable Data Consumption checkbox. |  |
| 1. Copy over the Database User Name, Host Name, Port and Password to for example notepad. In case you lost the password later you can always return to this dialog and request a new password. |  |
| 1. Select the newly created database user and hit the Open Database Explorer button on the right. |  |
| 1. In the SAP HANA Database Explorer (you can also use any other 3rd party SQL tool instead) hit the Add Database button. |  |
| 1. Select the Database Type SAP HANA Database and enter the host, Port number, User and Password which you copied over from the other dialog in SAP Data Warehouse Cloud. Make sure to check the checkboxes Save password and Connect to the database… Make sure to un-check the Verify the server’s certificate… checkbox. |  |
| 1. Select the value help button. |  |
| 1. Choose the schema matching your space name, for example ANA365\_001. |  |
| 1. Expand the database you just added and scroll down to Views. |  |
| 1. Select Views and select your view Sales\_Data from the list of available views below. |  |
| 1. A new tab opens displaying the columns of this view like you created it in SAP Data Warehouse Cloud. Hit the Open Data button in the top right corner to preview the available data. |  |
| 1. The same data is displayed as in SAP Data Warehouse Cloud. The data is fetched live from the view you created in the Data Builder in your space. |  |
| 1. Congratulations! | You now know how to consume data models within SAP Data Warehouse Cloud using the Story Builder or externally via the Open SQL Schema using Database Users enabled for consumption and any external 3rd party SQL client.  Let’s continue with the last section of this hands-on! We will now check out how we can replicate data to SAP Data Warehouse Cloud using the Remote Table replication and View Materialization. |
| 1. In SAP Data Warehouse Cloud navigate to the Data Integration Monitor. |  |
| 1. For any remote table you create in your space in SAP Data Warehouse Cloud you can control whether the data access is remote (no data is stored permanently in your space, but always accessed live during query runtime) or replicated (real-time or snapshot). Using the Table Replication menu in the top right corner you can enable snapshot or realtime replication if available. |  |
| 1. Select the line for the Product\_Attributes table and then select Load New Snapshot to replicate the remote table Product\_Attributes to your space for the sake of demonstrating the Snapshot Schedule option. For specific data sets you can also enable Real-Time Access if required. |  |
| 1. Refresh the page and wait until the Status changes to Available. |  |
| 1. For snapshot replications you can also create a schedule. Creating a schedule allows you to automatically create new snapshots based on the provided frequency. Go ahead and create a Snapshot Schedule for this table. |  |
| 1. Clicking on the little arrow icon on the right takes you to the replication monitor logs for the data replication tasks for the selected table. |  |
| 1. The Logs screen gives you an overview of all past and currently running replications. |  |
| 1. Replicating a remote table always replicates all the data from this table 1:1. This might not be needed if you apply filters in views that use this table and therefore only a require a subset of the actual data of this remote table. Go back to the Remote Table Monitor and change the Data Access back to Remote. |  |
| 1. To only persist the result of a modeled view, select the View Persistency Monitor. Hit the Add button on the right create a new view persistency. |  |
| 1. From the dialog select the Products view. |  |
| 1. After confirming the dialog, you can see that a new view persistence was created and the status is Loading. |  |
| 1. Hit Refresh until the status changes to Available. |  |
| 1. Congratulations! | You made it! You finished the exercise and experienced all the different aspects of data integration in SAP Data Warehouse Cloud. As a review, during the session you you learned about, how to create federated and replicated connections to SAP & non-SAP, cloud and on-premises data sources, how to load data into SAP Data Warehouse Cloud using external SQL clients, how to upload CSV files, how to model ETL processes to extract data from external sources into SAP Data Warehouse Cloud, how to consume & visualize data models using the SAP Analytics Cloud story builder and external SQL clients, … hope you enjoyed it! |