SACM21

SAP Analytics Cloud: Data Modeling

PARTICIPANT HANDBOOK
INSTRUCTOR-LED TRAINING

Course Version: 34
Course Duration: 2 Day(s)
Material Number: 50161716

SAP Copyrights, Trademarks and Disclaimers

Typographic Conventions

American English is the standard used in this handbook.

The following typographic conventions are also used.

This information is displayed in the instructor's presentation	
Demonstration	
Procedure	
Warning or Caution	
Hint	
Related or Additional Information	
Facilitated Discussion	
User interface control	*Example text*
Window title	*Example text*

Contents

Course Overview

TARGET AUDIENCE

This course is intended for the following audiences:

- Technology Consultant
- Industry / Business Analyst Consultant
- Business User
- Business Analyst

UNIT 1 Explaining the basics of data structures in SAP Analytics Cloud

UNIT OBJECTIVES

- Describe the foundational concepts for data modeling
- Explore the different model structures
- Describe the connection types and data sources
- Explain the types of datasets and the difference between datasets and models
- Create a dataset

Unit 1
Lesson 1

Describing data modeling basics

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Describe the foundational concepts for data modeling

Implement SAP Analytics Cloud modeling

SAP Analytics Cloud combines Business Intelligence, augmented and predictive analytics, and enterprise planning capabilities in one product so everyone has the capabilities they need to make fast, confident decisions. In order to do this, you need information from SAP Analytics Cloud stories that are based on real time or import models. You can use off-shelf-content or you can create your own but in order to so, you need modeling skills.

Before we get into the modeling concepts, let's review the business scenario that we are using for the course in the *SAP Analytics Cloud Business Scenario* figure below. In the next paragraph, let's consider the situation for the business user.

Business scenario

You have recently joined a project team that is implementing HR, sales and finance analytics using SAP Analytics Cloud. You are new to modeling in SAP Analytics Cloud and so you are interested in how the business requirements can be met and what options are available. You are taking direction from an experienced system architect on the project.

Preliminary scope:

- Real time sales and HR data in stories so analytics can be up to date all of the time.
- Replicated finance data that is refreshed once per day. In other words, finance doesn't need real time data, they only need daily snap-shots.
- - The models will use public dimensions that will be shared among multiple models.
  - Import models will be used to store the transaction data.

Stories access data from its data acquisition layer which is made up of datasets, import models and live data models.

Note:
Real time data resides in the source system and is viewed in SAP Analytcs Cloud (SAC). Replicated data resides in both the source system and in SAC

Use case: Implement sales, HR, and finance stories for analytics and planning

- Project phases:
  - Phase I - Implement analytics
  - Phase II - Implement planning
- Business requirements for data access:
  - Replicated data is needed for finance
  - Real time data is needed for sales and HR

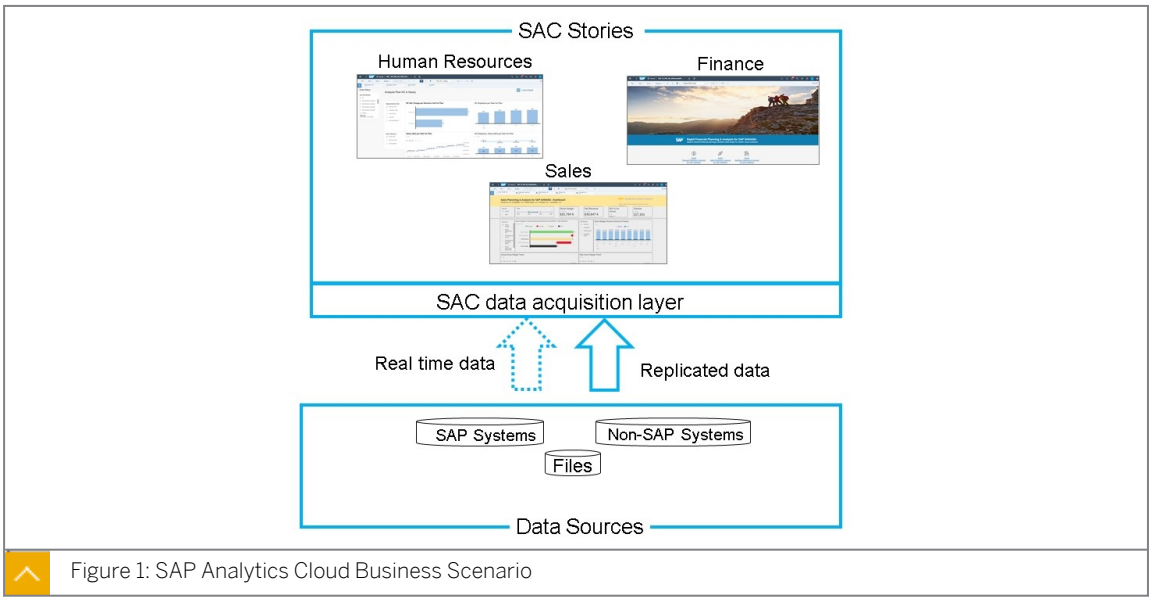


Figure 1: SAP Analytics Cloud Business Scenario

In this course, we'll address the modeling aspects to implement an analytics solution. In the business requirements listed below, you will get a preview of the kinds of questions that the course addresses. That way, you will have a better idea regarding what questions to ask on your project.

Business requirements

As always, the gathering of business requirements will drive the implementation process. The business requirements are normally driven by the stories that are needed for the business users. As the components are set up and the stories and applications are built, it is important to do stress testing as soon as possible particularly where the stories are complex and have high data volumes.

In addition, keep in mind how much your data will grow over time since this may affect performance.

Based on the business scenario, what modeling decisions will you be faced with?

- Data access:
  - What source systems is the data coming from?
  - Is the data accessed via live or import connections?
- For replicated data scenarios using import connections:

  - Will analytic and/or planning models be used?
  - What dimensions and measures are needed in the models?
  - What dimension properties and hierarchies are needed?
  - Will public or private dimensions be used?
  - What dimensions are needed for the sales vs. finance models?
  - Is there a need to do currency translation?
  - What are the time requirements e.g., weekly, monthly, custom hierarchies?
  - Where are calculations needed?
  - Are geo maps required?
  - What are the data access control requirements?
- For real time data scenarios using live connections:
  - Will live models will be used?
  - What are the touch points to various SAP systems?

Note:
Live data can be accessed via live models or with the data analyzer without a model.

Note:
Geo mapping is available for both live and replicated data scenarios, however, in this course we are focusing on the replicated use case.

LESSON SUMMARY
You should now be able to:

- Describe the foundational concepts for data modeling

Discussing the difference between model structures

LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Explore the different model structures

Account-based models

There are two types of account-based models:

- Classic account model
- New model

Single measure with multiple accounts

"Account-based" concept

Available with:

- ✓ Classic account model
- ✓ New model

Product	Date	Account	SignedData
Tennis Shoes	Q2 2022	Discounts	$700000
Tennis Shoes	Q2 2022	Gross Sales	$-6000000
Tennis Shoes	Q2 2022	Quantity	40000
Tennis Shoes	Q2 2022	Cost of Sales	$2000000

Figure 2: Account-based models

In a classic account or new model, model values are stored in a single default measure, and you use the account structure to add calculations, specify units, set up an account hierarchy, and set aggregation types for all the data. For example, financial data is broken down by general ledger account (i.e., discounts and gross sales), which is used to describe the values in the measure column (i.e., signed data), which represents transaction data.

Measure-based models

There are two types of measure-based models:

- New model without accounts
- New model with accounts

Multiple measures with no account dimension

“Measure-based” concept

Available with:

✓ New model

Product	Date	Discounts	Gross Sales	Quantity	Cost of Sales
Tennis Shoes	Q2 2022	$700000	$6000000	40000	$200000

Figure 3: New model without accounts

The new model type exposes measures as single entities and lets you add and configure multiple measures with aggregation and units to fit your data. It adds plenty of flexibility: you can still match the structure of a classic account model by using a single measure and an account dimension, or you can remove the account dimension when it's not required for your use case.

Multiple measures with an account dimension

Available with:

✓ New model

Product	Date	Account	Local Amount	Group Amount	Quantity
Tennis Shoes	Q2 2022	Discounts	$700000	770000 Euro	2000
Tennis Shoes	Q2 2022	Gross Sales	$-6000000	$-6600000 Euro	2000
Tennis Shoes	Q2 2022	Cost of Sales	$4000000	$4400000 Euro	2000
Tennis Shoes	Q2 2022	Administration	$250000	$280000 Euro	

Figure 4: New model with accounts

The new type model is a best of both worlds. It has accounts but it can have multiple measures which is used to match most financial data models. In addition, new models support model-specific calculated and converted measures.

The main difference between the classic and new model types is how they handle measures.

Table 1: Model type comparison

Feature	Classic	New Model
Measures	1	1 or more
Converted measures	Not available	Supported
Calculated measures	Not available	Supported
Account dimension	Required	Optional
Analytic or planning	Both	Both

Benefits of the New Model

The New Model offers the following main benefits, both for analytic and planning use cases:

- Flexible model structure

Since both accounts and measures are available as structures for your data, you can display your data more precisely for a variety of use cases, such as:

- Accurate aggregation: Aggregating data over measures instead of accounts opens up some new options. For example, you can generate your balance sheet and profit and loss statement out of a single model.
- Explicit data types: Measures can be set up as integers or decimals. By setting different types of data for different measures like monetary amount, number of units, operating hours, an so on, you can avoid incorrect data; for example, using decimals when planning on headcount values.
- Disaggregation of data for both integer and decimal measures: With an integer measure, you can avoid decimals while still distributing the entire value

- Optional account dimension

  When you structure your data with measures, you can add your accounts to a generic dimension instead of an account dimension. This option allows you to avoid limitations on the account dimension.

- Improved calculations
  - Calculated measures in models: Since measure calculations can be added to your model, you can reuse them across different stories and analytic applications and also add them to some data action steps.
  - Calculations on numeric dimension properties: Using this feature, you can change account values from positive to negative based on dimension properties. For example, you can create calculations that show either positive or negative values for expenses, which makes it easier for different types of users like controllers or accountants to analyze and plan on the same data.

- Enhanced currency features
  - Base currencies and conversion measures in models: You can add multiple base currency measures to your model and then add currency conversion measures on top of them. This way, you can model and plan on multiple currencies like transaction, local, or group currencies. With currency conversions available in the model, you can use them in formulas as well as data action copy steps.
  - Planning across currencies: In tables, you can plan on any base measure or conversion measure, such as local, transaction, and group currencies, and instantly see the results across dependent currencies. And you can apply currency conversion while copying data between base measures with a data action.

- Data integration

  The measure model matches the structure of data from several other SAP systems more closely, including SAP S/4 HANA, SAP BW, and SAP IBP. Data integration is generally faster and requires fewer transforms because you can import data from multiple measures directly instead of turning them into account members.

- More clear terminology for charts and tables

  By removing non-financial data from members of the account dimension, you can make data in charts and tables easier to understand for your non-accounting/financial viewers and content creators.

- **Flexible model structure**

  Since both accounts and measures are available as structures for your data, you can display your data more precisely for a variety of use cases, such as:

  - Accurate aggregation: Aggregating data over measures instead of accounts opens up some new options. For example, you can generate your balance sheet and profit and loss statement out of a single model.
  - Explicit data types: Measures can be set up as integers or decimals. By setting different types of data for different measures like monetary amount, number of units, operating hours, and so on, you can avoid incorrect data; for example, using decimals when planning on headcount values.
  - Disaggregation of data for both integer and decimal measures: With an integer measure, you can avoid decimals while still distributing the entire value

Analytic models

Analytic models:

- Use case: Store transaction data that is used to make business decisions
- Required dimensions: None (e.g. no date dimension is required)
- Data: Populated during the import process and is read only.

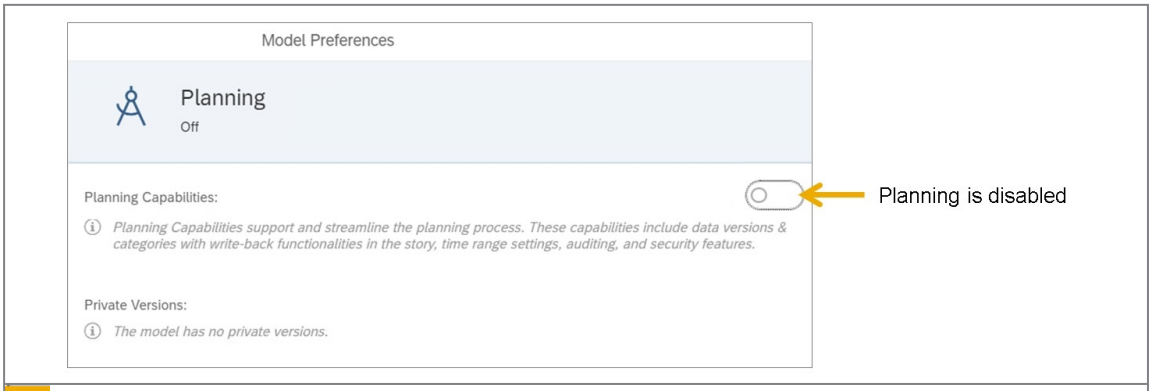


Figure 5: Analytic model concept

Analytic models don't have any required dimension types. For example, a date dimension is not required.

Analytic models are populated via the import process. After the data import, the data is read only.

Key question: Why would you use analytic models instead of live data models?

You would use an analytic model when:

- You need data in a story that is refreshed daily (i.e., not real time)
- Source system performance issues preclude the use of live models
- You need to transform data during the import process
- You need to physically combine data from multiple source systems
- You need model-dependent calculated measures

Planning models

Planning models:

- Use case: Story data that is used for planning purposes
- Required dimensions: Version and time
- Data: Populated during the import process

Figure 6: Planning model concept

Why use planning models?

- You are implementing planning in SAP Analytics Cloud.
- Planning models are required in this scenario.

In a typical workflow, after planning models are created, actual data is imported. The actual data is typically copied to a plan version and then the data is adjusted to account for expected changes in the planning time frame.

Note:
The version dimension is used to distinguish between actual vs. plan data.

New type models do not require account dimensions but classic models do. In addition, planning models require a date and version dimension.

LESSON SUMMARY

You should now be able to:

- Explore the different model structures

Unit 1
Lesson 3

Exploring the prerequisites for creating models in SAP Analytics Cloud

LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Describe the connection types and data sources

Connection Types

SAP Analytics Cloud is a public Software-as-a-Service (SaaS) that enables access to on-premise and cloud data sources. SAP Analytics Cloud provides live connection (online) and data import (batch) connectivity – two ways of accessing your data located anywhere in your information-system landscape.

The figure *Connection Types* illustrates the connection types and data connections environments.

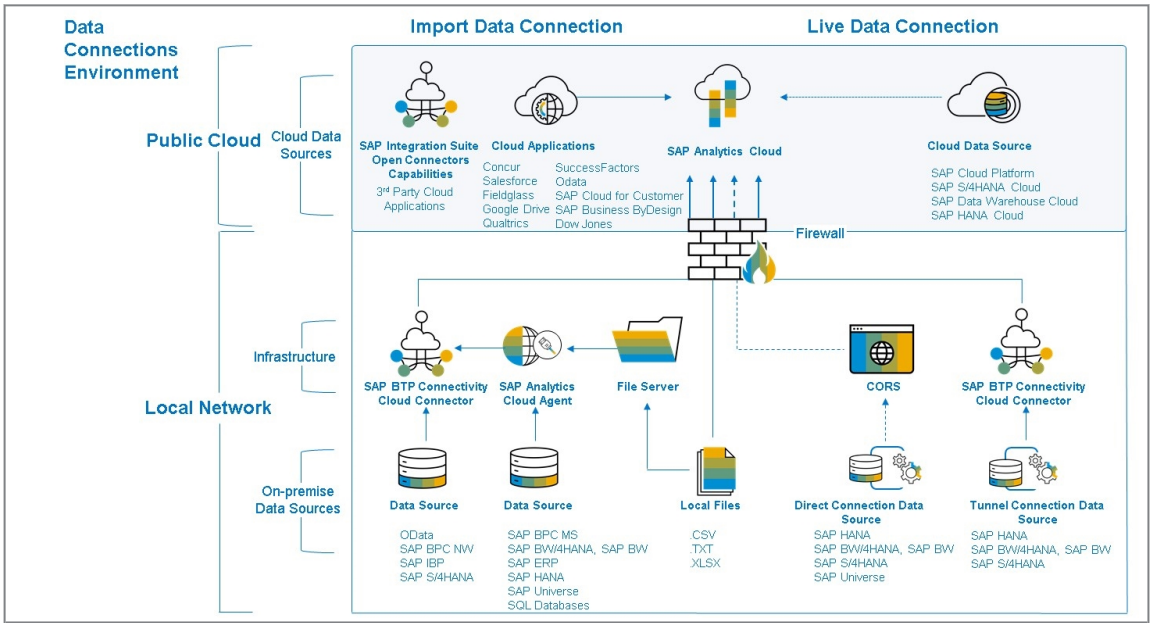


Figure 7: Connection Types

The SAP BTP Connectivity Cloud Connector serves as the link between SAP Analytics Cloud and existing on-premise systems. The SAP BTP Connectivity Cloud Connector combines an easy setup with a clear configuration of the systems that are exposed to SAP Analytics Cloud. In addition, you can control the resources available for the cloud applications in those systems. Thus, you can benefit from your existing assets without exposing the whole internal landscape to SAP Analytics Cloud.

The SAP Analytics Cloud Agent is an on-premise data connectivity component that is used to:

- Import data connections from SAP Business Planning and Consolidation, version for Microsoft Platform (BPC MS).
- Import data connections from SAP Business Warehouse (BW).
- Import data connections from SAP Universes.
- Import data connections from SAP ERP.
- Import data connections to an SQL database.
- Import data from a file server.

For more information on the cloud agent, see:https://help.sap.com/viewer/00f68c2e08b941f081002fd3691d86a7/release/en-US/7cb6ffb38c294a5c871d6cc6ad5b1b36.html

Depending on your environment, there are different requirements for establishing both live and import data connections to SAP Analytics Cloud. SAP recommends creating your live data connections using a direct connection type (Cross-Origin Resource Sharing (CORS)). Direct connectivity does not require any additional hardware. It is easy to set up and provides superior performance.

Live Connections

With live connections, you can create models from data sources in on-premise or cloud systems, build stories based on those models, and perform online analysis without storing your data in the cloud. This feature allows SAP Analytics Cloud to be used in scenarios where data cannot be stored into the cloud for security or privacy reasons, or your data already exists on a different cloud system.

Import Connections

Data is imported (copied) to SAP Analytics Cloud, and changes made to the data in the source system do not affect the imported data. Setup of SAP BTP Connectivity Cloud Connector and SAP Analytics Cloud Agent is required when creating an import data connection to system types such as:

- SAP Business Warehouse (BW)
- SAP Business Planning and Consolidation (BPC)
- SAP BusinessObjects Business Intelligence platform Universe (UNX)
- SAP ERP
- SQL Database
- SAP SuccessFactors
- WorkforceAnalytics
- OData
- SAP Concur
- Salesforce.com (SFDC)
- SAP Fieldglass

- Google Drive
- Google BigQuery
- File server

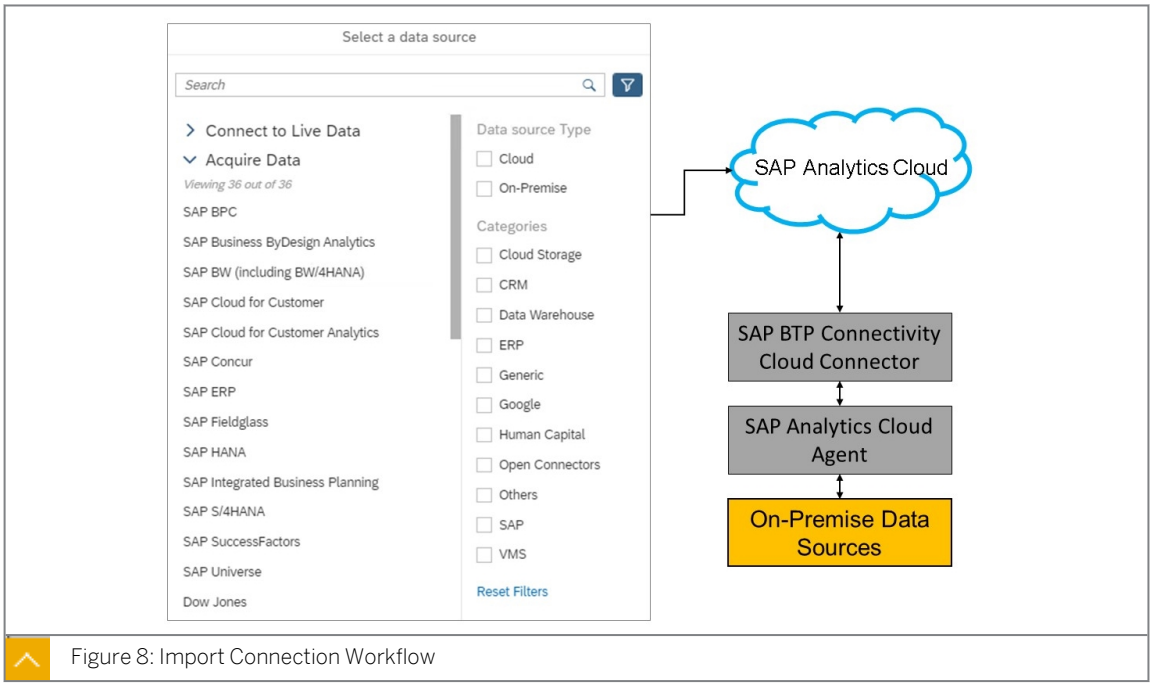


Figure 8: Import Connection Workflow

Customers need to decide which connection type to set up, according to their own needs. As you evaluate which connection type to use, consider the following criteria:

- Functional needs
- Data privacy constraints
- Data volume constraints

It is also important to review the system requirements and technical prerequisites, and to check whether your landscape is compliant with what is supported for your version and connection type.

For more information, see: https://help.sap.com/viewer/00f68c2e08b941f081002fd3691d86a7/release/en-US/11b4e5ff76eb4747bc255d7037be1f01.html

Project management is also required as maintaining connectivity settings cannot be successful as a one-person task. Settings follow a strict process where different stakeholders must be engaged and must deliver their own expertise in their respective areas of responsibility.

Connection Screen

The figure *Connection Creation in SAP Analytics Cloud* shows the connection screen in SAP Analytics Cloud.

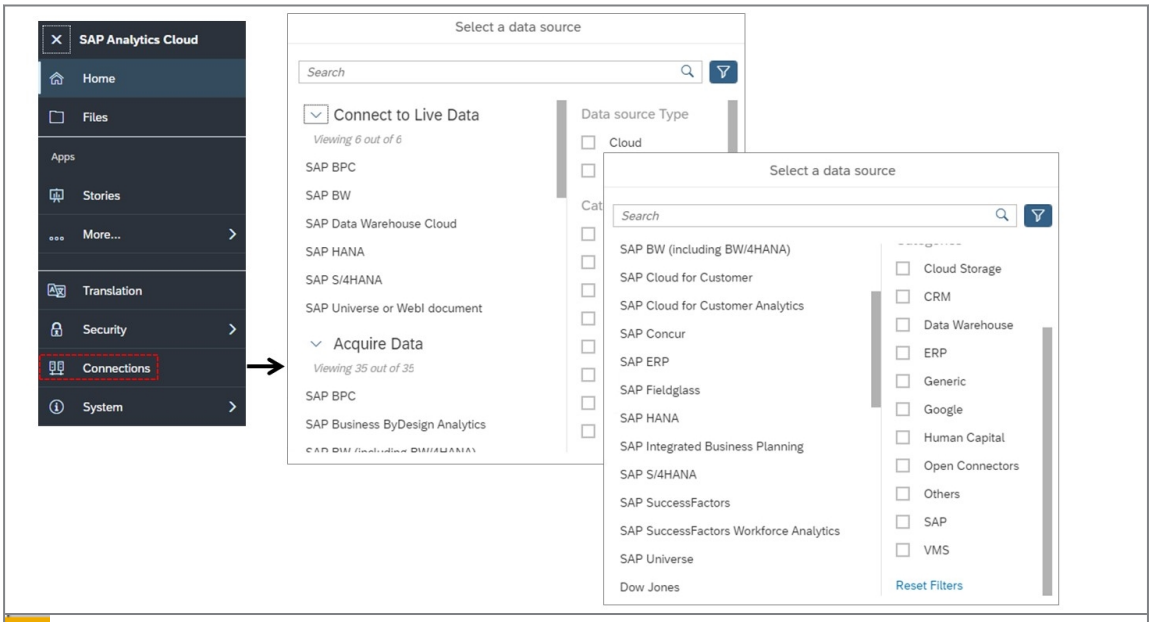


Figure 9: Connection Creation in SAP Analytics Cloud

On the Connection screen, you create and maintain datasource system connections. Connections are categorized as either a live connection or an acquired data connection. It is also possible to filter the list of available connections by datasource type or by category.

Data sources for import models

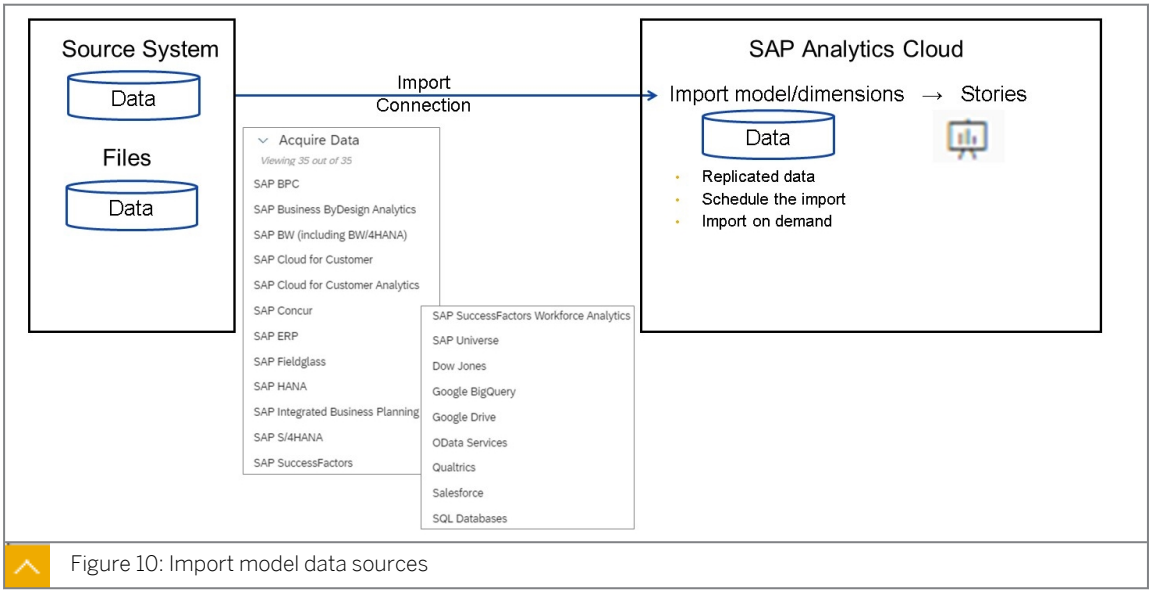


Figure 10: Import model data sources

You can create connections to data source systems to allow data acquisition by SAP Analytics Cloud. Data is imported (copied) to SAP Analytics Cloud, and changes made to the data in the source system don't affect the imported data. Imports from SAP sources systems can be scheduled. Import from files cannot be scheduled.

Import models support model-based calculations as well as dimension formulas.

When importing into dimensions and import models, the data can be transformed. Data access is controlled by SAP Analytics Cloud security.

Data sources for live models

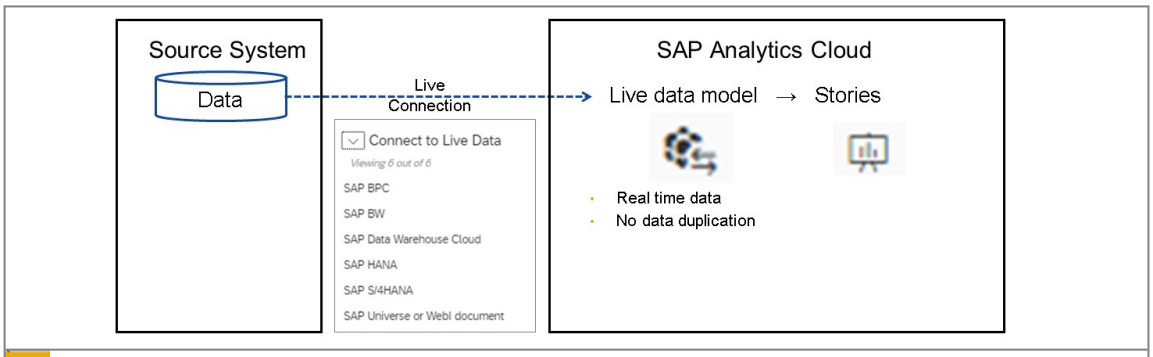


Figure 11: Live model data sources

In SAP Analytics Cloud, you can create models from data sources in on-premise or cloud systems, build stories based on those models, and perform online analysis without data replication. This feature allows SAP Analytics Cloud to be used in scenarios where data cannot be moved into the cloud for security or privacy reasons, or your data already exists on a different cloud system. The data from the source system is not transformed.

Live data models don't support model-based calculations but story calculations are available.

Data access is controlled by source system security, not SAP Analytics Cloud security.

LESSON SUMMARY

You should now be able to:

- Describe the connection types and data sources

Unit 1
Lesson 4

Introducing datasets

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Explain the types of datasets and the difference between datasets and models
- Create a dataset

Datasets

A dataset is a simple collection of data, usually presented in a table. You can use a dataset as the basis for your story, and as a data source for Smart Predict.

Dataset are first choice when you want to create story/ visualization quickly and do not want to get into structure definition, during data processing, or when development does not demand IT governance.

For the most part, datasets acquire data via import connections. The only exception is that a dataset can be built based on a live connection to a HANA data repository.

Table 2: Datasets vs. import models

Characteristics	Datasets	Planning Models	Analytic Models
Purpose	Ad-hoc/Smart predict	Planning	BI
Scheduled imports available	No	Yes	Yes
Standalone or embedded	Both	Standalone	Standalone
Table	1 data table	Dimensions + data table	Dimensions + data table
Calculated measures	No	Yes	Yes

Public and private datasets

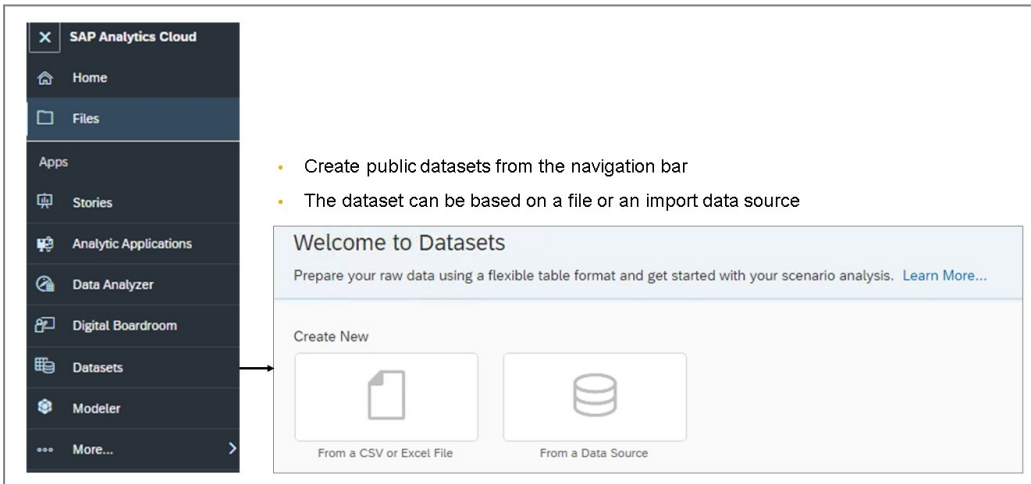


Figure 12: Dataset types

Public datasets:

- Public datasets are standalone
- Can be used in multiple stories
- Can be used for smart predict
- Created from the main menu

Private datasets:

- Dataset that is embedded into a story
- Cannot be used for smart predict
- Created from an existing or new story

Embedded datasets: When you create a story and import data from a file or other data source, but not from an existing saved model or dataset, that data is saved as an embedded dataset (also called a private dataset) within the story, and this dataset doesn't appear in the Files list. However, if you want others to be able to use this dataset, you can convert it to a public dataset.

> Note:
> A shareable live dataset can be created for SAP HANA. Live datasets can also be created from a story based on SAP Datasphere.

LESSON SUMMARY

You should now be able to:

- Explain the types of datasets and the difference between datasets and models
- Create a dataset

Learning Assessment

1. Geo mapping is available for both live and replicated data scenarios.

   Determine whether this statement is true or false.

   ☐ True

   ☐ False

2. A typical account-based model only has one measure.

   Determine whether this statement is true or false.

   ☐ True

   ☐ False

3. A planning new-type model must have an account dimension.

   Determine whether this statement is true or false.

   ☐ True

   ☐ False

4. Live connections require the cloud agent.

   Determine whether this statement is true or false.

   ☐ True

   ☐ False

5. Live models support model-specific calculated measures.

   Determine whether this statement is true or false.

   ☐ True

   ☐ False

6. Datasets require an account dimension.

   Determine whether this statement is true or false.

   - [ ] True
   - [ ] False

7. Private datasets are user specific.

   Determine whether this statement is true or false.

   - [ ] True
   - [ ] False

Learning Assessment - Answers

1. Geo mapping is available for both live and replicated data scenarios.

   Determine whether this statement is true or false.

   [X] True

   [ ] False

   Correct. Geo mapping is available for both live and replicated data scenarios.

2. A typical account-based model only has one measure.

   Determine whether this statement is true or false.

   [X] True

   [ ] False

   Correct. A typical account-based model only has one measure.

3. A planning new-type model must have an account dimension.

   Determine whether this statement is true or false.

   [ ] True

   [X] False

   Correct. A planning new-type model does not require an account dimension.

4. Live connections require the cloud agent.

   Determine whether this statement is true or false.

   [ ] True

   [X] False

   Correct. Live connections do not use the cloud agent.

5. Live models support model-specific calculated measures.

   Determine whether this statement is true or false.

   ☐ True

   ☒ False

   Correct. Live models do not support model-specific calculated measures.

6. Datasets require an account dimension.

   Determine whether this statement is true or false.

   ☐ True

   ☒ False

   Correct. Datasets don't have dimensions.

7. Private datasets are user specific.

   Determine whether this statement is true or false.

   ☐ True

   ☒ False

   Correct. Private datasets are story specific.

UNIT 2 Designing and creating dimensions

UNIT OBJECTIVES

- Explain private and public dimensions and the difference between dimensions and measures
- Explain account, organization, generic, data and version data types
- Explain the unique properties for dimension types
- Explain the prerequisites required for creating hierarchies
- Create a dimension and hierarchy
- Import and wrangle data in a dimension

Defining dimensions

LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Explain private and public dimensions and the difference between dimensions and measures

Dimensions and measures

In SAP Analytics Cloud, dimensions represent master data in general. Product, cost center, and employee are all examples of dimensions. A model can have several dimensions.

On the other hand, a measure represents transactional data (quantitative). In a classic account model, model values are stored in a single default measure, and you use the account structure to add calculations, specify units, and set aggregation types for all the data. In the new model type, you can add and configure multiple measures with aggregation and units to fit your data.

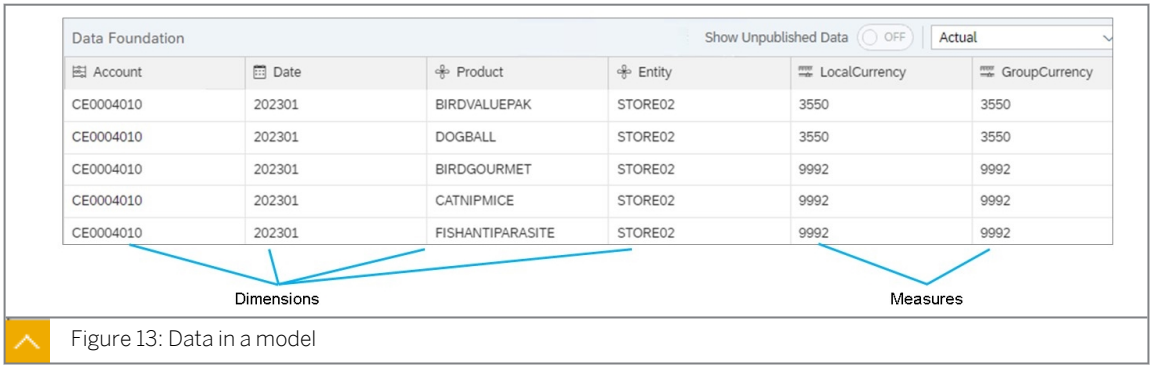


Figure 13: Data in a model

Measures are quantitative:

- Examples: amount, quantity, price, number of people
- Import models store measure values (numbers)
- Measures contain settings for: aggregation and currency
- Restricted measures are refreshed on the fly:
  - Examples: Actual Revenue, Plan Revenue, Expense for 2024
  - Restricted measure results are not stored
- Calculated measures are also refreshed on the fly:
  - Examples: Gross margin %, Calculated Price

- Calculated measure results are not stored

Note:
In the figure above (Data in a model), the local currency measure refers to an amount in the currency of entity (company). For example, a US company has labor expense of 3550 in USD. The group currency measure is used to store the values in the corporate currency. For example, if the US company has a German parent, the group currency in Euros.

Note:
Semantically, an account may be referred to as a measure because it fulfills a similar purpose. However, accounts are not measures, they are part of master data.

Dimensions are qualitative:

- Dimensions represent master data
- Examples: Cost center, account, product
- How do accounts relate to measures?
  - Accounts are used to describe measure values
  - In the data table above, the CE0004010 labor account identifies the measure mount value of $250,000 as a labor expense

Private and public dimensions

About dimensions

In SAP Analytics Cloud, dimensions represent master data in general. For example, product, cost center, and employee are all examples of dimensions. A model can have any number of dimensions.

On the other hand, a *Measure* represents transactional data (quantitative). In a classic account model, model values are stored in a single default measure, and you use the account structure to add calculations, specify units, and set aggregation types for all the data. In the new model type, you can add and configure multiple measures with aggregation and units to fit your data.

Public vs. public dimensions

Public dimensions

- Main use case
- Can be shared by several models
- Created from the model & the main menu
- When the model is copied or deleted, public dimensions are not affected
- Update: Manual | Dimension import
- Schedule imports from SAP source systems

Private dimensions:

- Use case: ad hoc | infrequently used
- Model specific
- Create from the model
- When the model is copied or deleted, the private dimensions are as well Update: Manual | During model import

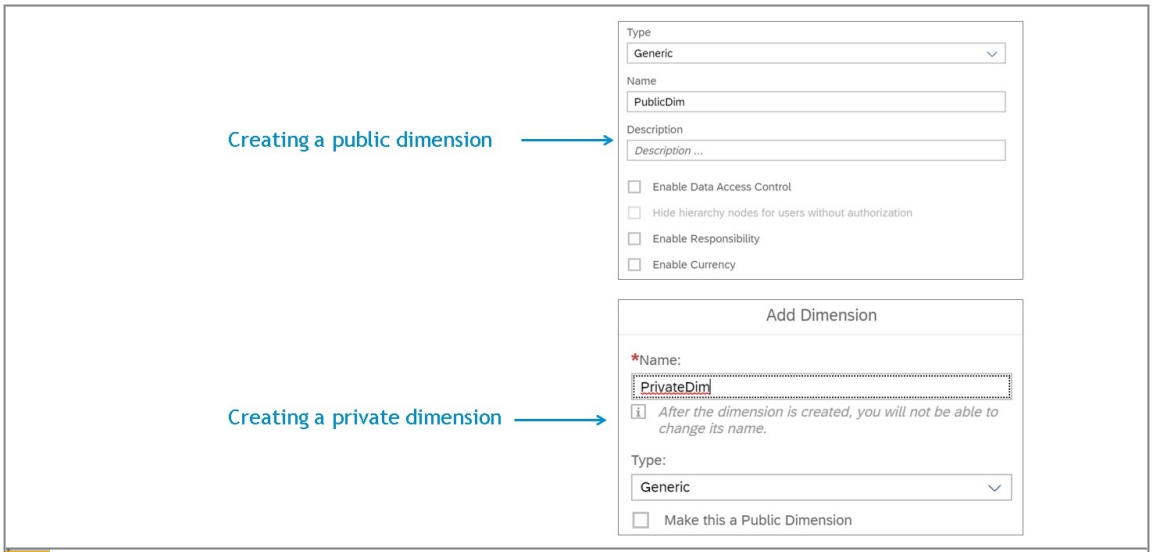


Figure 14: SAP Analytics Cloud Dimensions

Public dimensions are typically used for master data that needs to be shared among multiple models. Examples are cost center, company, and accounts. Public dimensions provide a centralized master data concept in SAP Analytics Cloud. Public dimensions provide a centralized master data concept in SAP Analytics Cloud.

Private dimensions may not be relevant on a customer-specific implementation but they can be used in a design mock-up scenario if needed.

Public dimensions have a data management workspace that is used to import from files or source systems. Private dimensions do not have a data management workspace.

LESSON SUMMARY

You should now be able to:

- Explain private and public dimensions and the difference between dimensions and measures

Describing the dimension types

LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Explain account, organization, generic, data and version data types

Dimension types

SAP Analytics Cloud dimension types

Dimensions are qualified as a specific type. Depending on the type selected, you can configure various properties and possibly create hierarchies for the dimension.

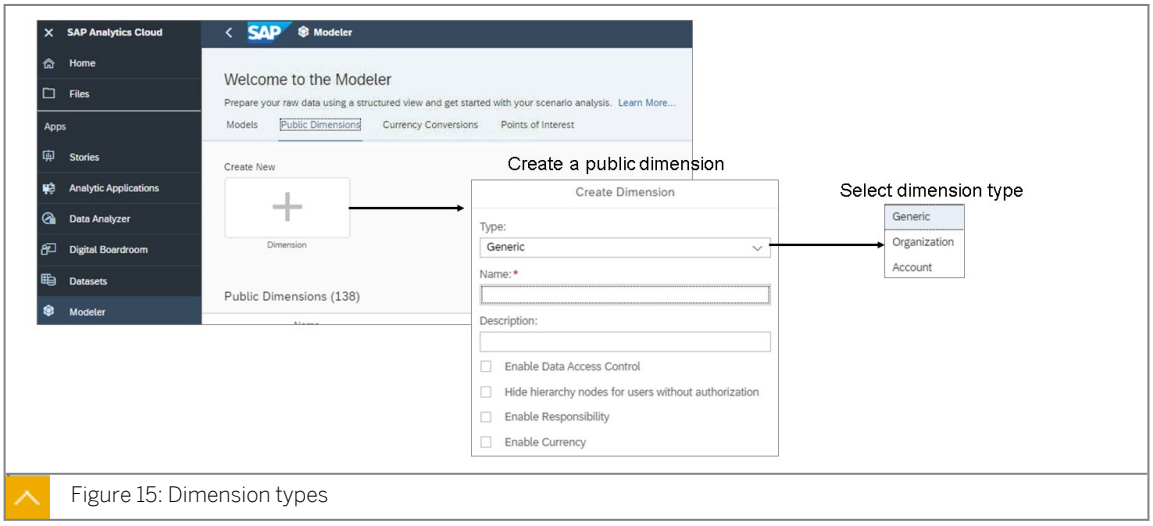


Figure 15: Dimension types

When creating a public dimension, you can select generic, account, or organization type. When creating a planning model, for example, the system adds version and date automatically.

In the following paragraphs, there is a summary of key points for each dimension type.

Account dimension type

- A dimension with financial accounts or others, such as kilowatt hours / number of people
- Properties: there are several system generated properties and more can be added.
- The default hierarchy is system generated, and more may be added
- Mandatory in a classic account model but optional in the new model type
- Optional in a mode

- Only 1 per model

Organization dimension type

- Represents an organizational structure, such as cost centers, profit centers, and business units
- Properties: currency and person responsible are system generated. More can be added
- Multiple hierarchies can be added
- Optional in a model
- Only 1 per model

Generic dimension type

- Properties: added as needed
- Multiple hierarchies can be added
- More than 1 per model allowed

Date dimension type

- A built-in dimension that defines the start and end dates of the model's timeline and specifies the granularity of time used in the model (years, quarters, months, weeks, or days)
- More than 1 time dimensions allowed in a model, Helpful if you need to compare dates for different scenarios, such as Order vs Shipping date
- The default hierarchy is system generated based on calendar time but can be changed to fiscal time
- 4-4-5 | 4-5-4 | 5-4-4 | 13x4 week patterns available (for new model types)
- 53 week support (for new model types)
- Custom properties and hierarchies available (for new model types)

Version dimension type

- A built-in dimension that defines the data versions: Plan, Actual, Forecast
- Only 1 per model

LESSON SUMMARY
You should now be able to:

- Explain account, organization, generic, data and version data types

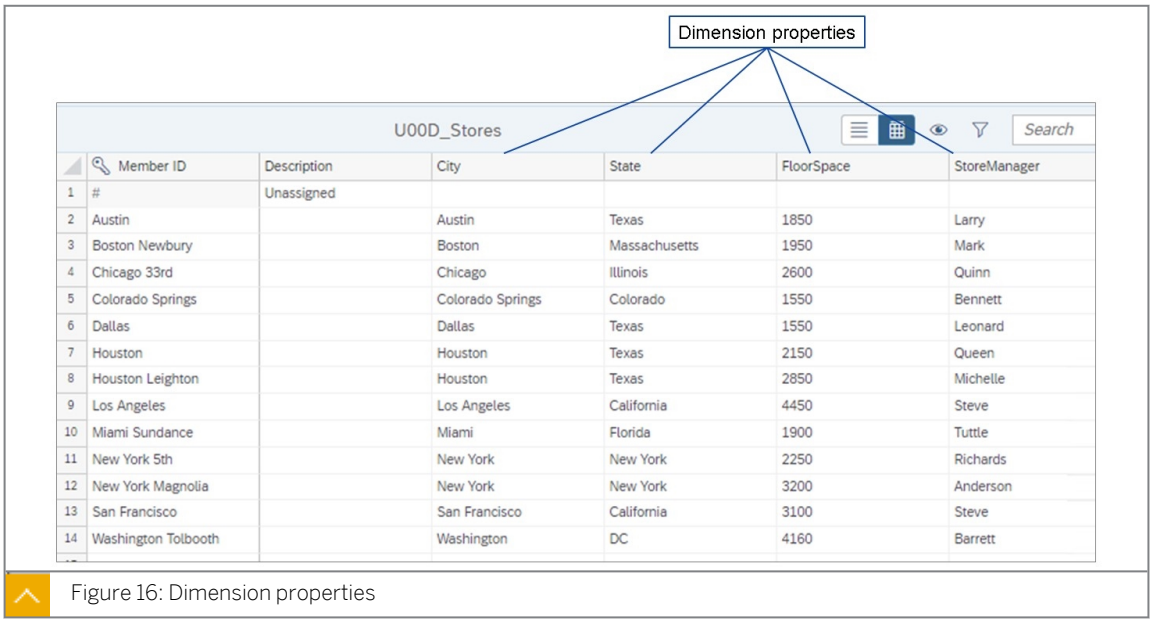
Describing properties

LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Explain the unique properties for dimension types

Dimension properties



	Member ID	Description	City	State	FloorSpace	StoreManager
1	#	Unassigned				
2	Austin		Austin	Texas	1850	Larry
3	Boston Newbury		Boston	Massachusetts	1950	Mark
4	Chicago 33rd		Chicago	Illinois	2600	Quinn
5	Colorado Springs		Colorado Springs	Colorado	1550	Bennett
6	Dallas		Dallas	Texas	1550	Leonard
7	Houston		Houston	Texas	2150	Queen
8	Houston Leighton		Houston	Texas	2850	Michelle
9	Los Angeles		Los Angeles	California	4450	Steve
10	Miami Sundance		Miami	Florida	1900	Tuttle
11	New York 5th		New York	New York	2250	Richards
12	New York Magnolia		New York	New York	3200	Anderson
13	San Francisco		San Francisco	California	3100	Steve
14	Washington Tolbooth		Washington	DC	4160	Barrett

Figure 16: Dimension properties

Note:
Member ID, description, and hierarchy appear as a columns in the member sheet, but they are not properties.

Dimension properties have many uses in SAP Analytics Cloud.

Here are a few examples:

- Sort and filter: in a story, filter companies based on their currency property
- Reporting: display the material type property for materials in a story
- Currency translation: translate average vs. end-of-month rates via the account dimension rate type property

The Account dimension type includes several unique properties:

- Account Type is tied to a feature that can reverse the sign. For example, revenue is stored as a negative value but viewed as a positive value.
- Rate Type is used for currency conversion.
- Units and currencies include pieces, gallons, or USD, EUR, and more.
- Formula is used to calculate values such as gross margin %.
- Aggregation, for example, prevent aggregation for an account that is used to store prices.

> Note:
> There are many aggregation types available in SAP Analytics Cloud: average, last, first, rank, sum. Aggregation behavior can be specified in account dimensions, calculated measures in stories as well as the New Model.

Unique properties for dimension types

Unique properties by dimension type

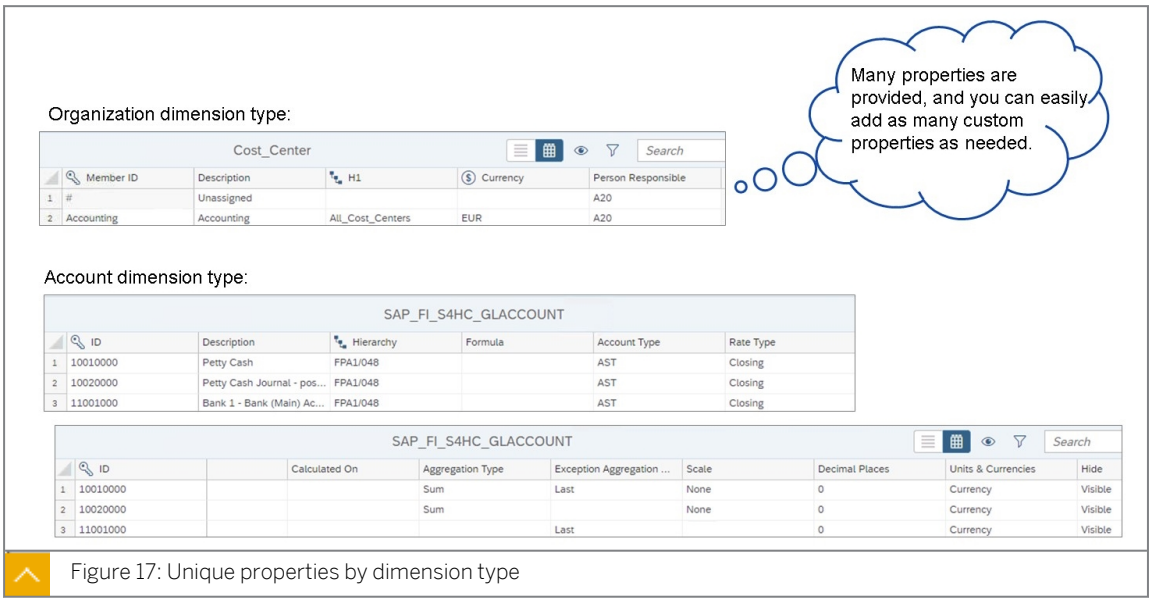


Figure 17: Unique properties by dimension type

The provided properties are unique for each dimension type:

- **Organization dimension type:**
  - On by default: currency and person responsible
  - Available: data access control, data locking ownership, hide unauthorized hierarchy nodes
- **Account dimension type:**
  - On by default: account type, formula, currency related, aggregation related
  - Available: data access control, data locking ownership, hide unauthorized hierarchy node
- **Date dimension type:**

- On by default: many date related such as day and week

- **Version dimension type:**
  - On by default: category
  - Available: data access control, data locking ownership
- **Generic dimension type:**
  - On by default: none
  - Available: currency, person responsible, data access control, data locking ownership, hide unauthorized hierarchy nodes

Reporting

Reporting with dimension properties

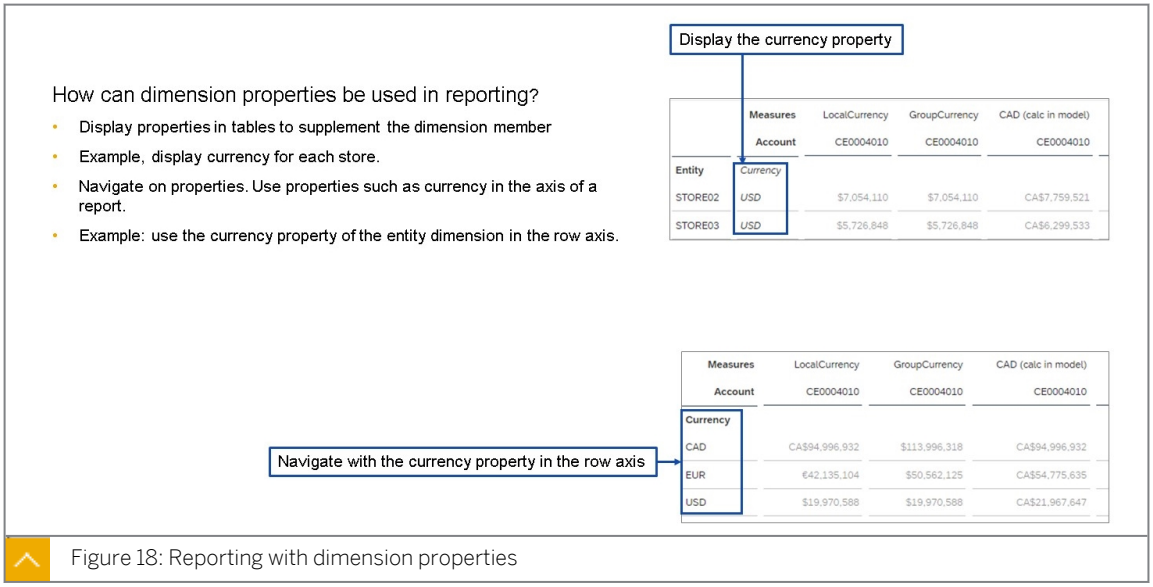


Figure 18: Reporting with dimension properties

In order to add a dimension property into a table for display purposes (using figure *Reporting with dimension properties* as an example):

- Hover on entity in the row axis
- Go to ...More → Properties → Select Currency

Sort and filter

Filter with dimension properties

In order to filter on a dimension property in a story, add an input control. In the input control, select the dimension that you want to filter and then select the property you want to filter on.

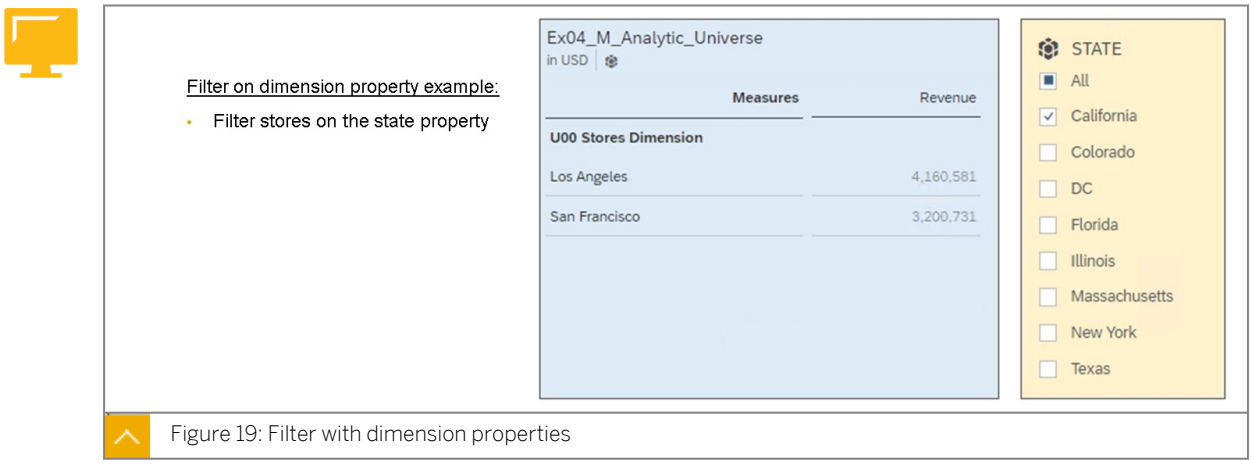


Figure 19: Filter with dimension properties

Sort on dimension properties

In order to sort on a dimension property (using figure *Sort on dimension properties* as an example):

- Add the property to the row axis (Choose +Add Dimension → Expand the stores dimension → Select the person responsible property)
- Right click on person responsible to sort it ascending or descending

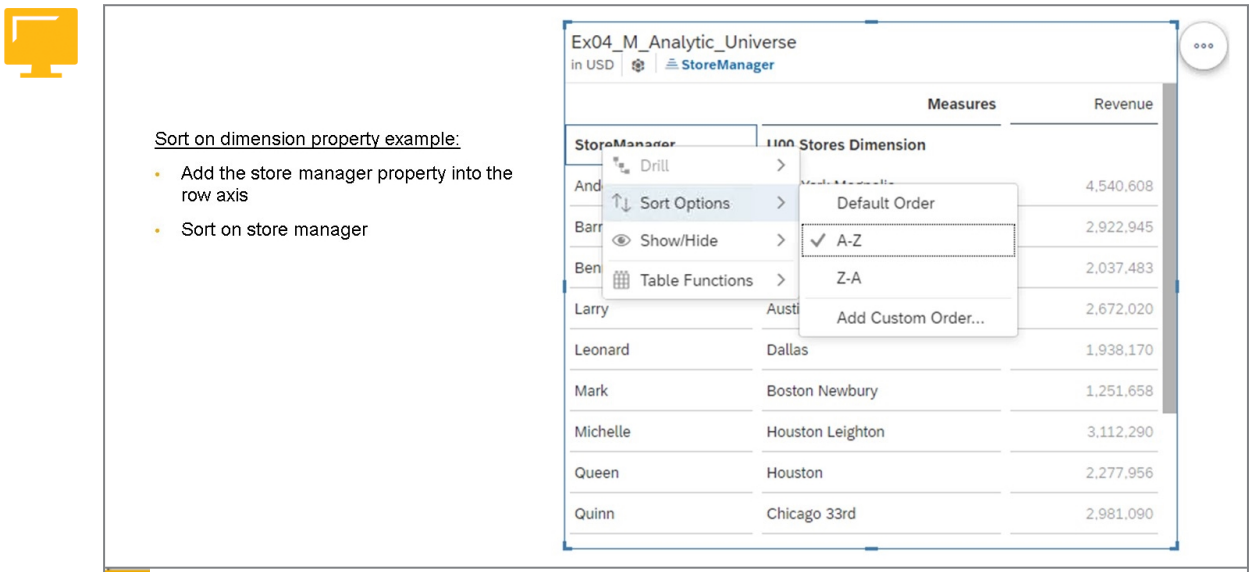


Figure 20: Sort on dimension properties

LESSON SUMMARY

You should now be able to:

- Explain the unique properties for dimension types

Unit 2
Lesson 4

Designing and creating hierarchies

LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Explain the prerequisites required for creating hierarchies
- Create a dimension and hierarchy

Hierarchies

A hierarchy is used to establish parent-child relationships within your data. For example, suppose you have sales data for the following:

- Worldwide
- Region
- Country
- State/Province
- City

A hierarchy allows you to organize these geographic areas into logical levels (also called nodes), from largest area to the most granular.

The hierarchy column in a dimension is used to store the parent-child values.

In a classic model, the account dimension has only a single hierarchy but multiple are allowed in new model types.

You can add hierarchies by selecting *+Add Hierarchy* on the toolbar. When you choose this option, a new hierarchy column is inserted into the dimension, and you must enter the name of the new hierarchy.

The hierarchy is visualized in the preview panel in the dimension. You can use drag and drop in the preview panel to arrange the members and build the relationships. If more than one hierarchy has been defined, you can select the one you want to display in the preview from the drop-down list.

Types of hierarchies

Two types of hierarchy are available:

- **Level-based hierarchy**

  A level-based hierarchy organizes multiple dimensions into levels, such as country, state, and city.

- **Parent-child hierarchy**

  A parent-child hierarchy organizes the members of a single dimension into a set of parent-child relationships.

Level-based hierarchy

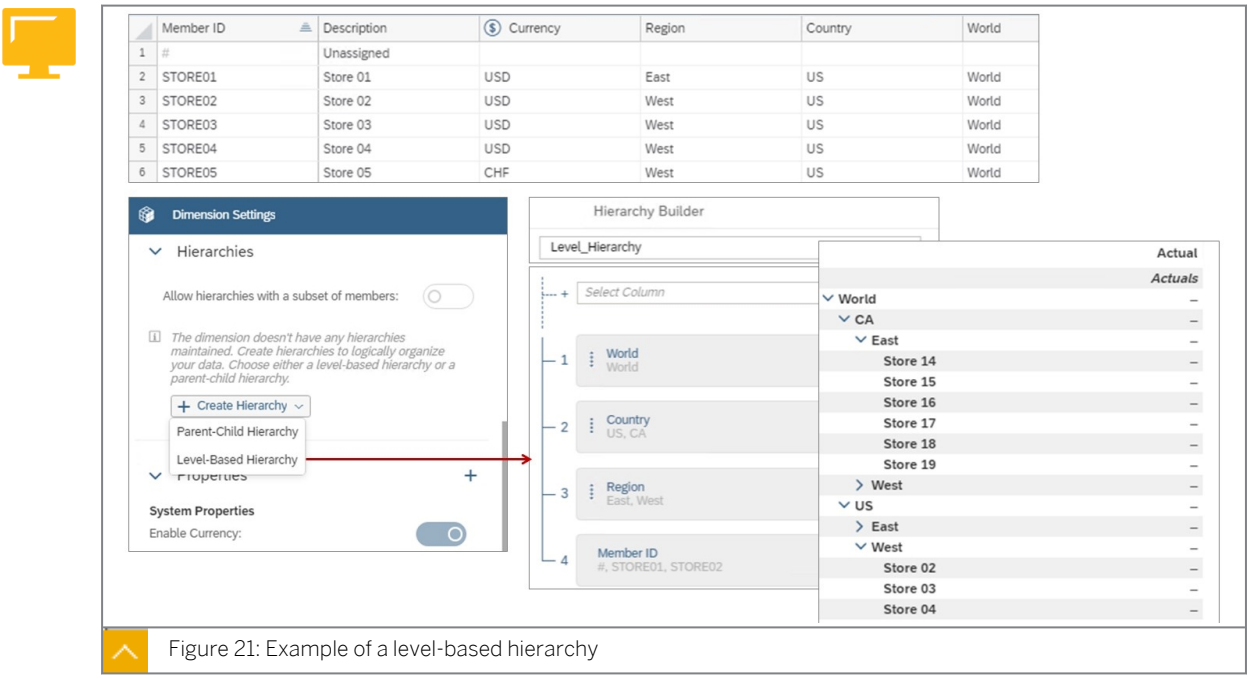


Figure 21: Example of a level-based hierarchy

The figure *Level-based hierarchy* shows dimensions with a level-based hierarchy (structure hierarchy).

Parent-child hierarchy

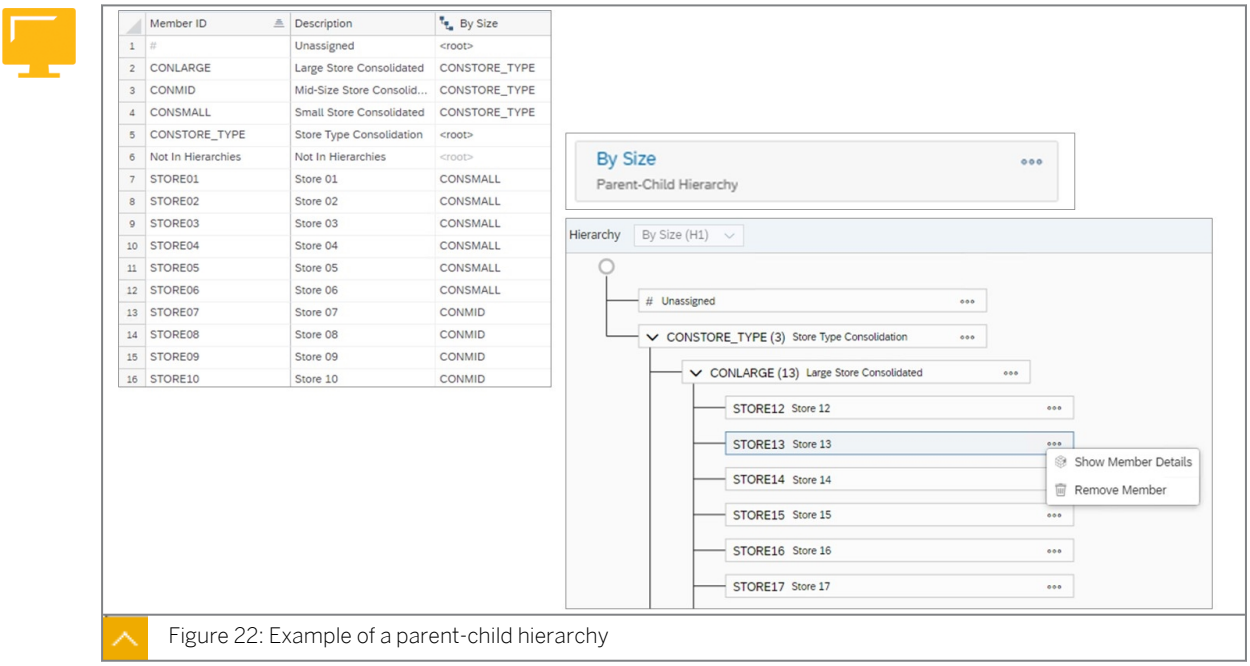


Figure 22: Example of a parent-child hierarchy

The figure *Parent-Child Hierarchy* shows dimensions with a parent-child hierarchy.

Parent-child hierarchies can be maintained in various ways:

- Manually in SAP Analytics Cloud
- Import from files
- Import from SAP systems, such as SAP BW and SAP S/4HANA

Note:
When working with parent-child hierarchies for a planning model, avoid situations where data can be booked directly to a parent node.

These situations include the following:

- Structuring two or more hierarchies so that a member is a leaf node in one hierarchy and a parent node in a different hierarchy.
- Updating a hierarchy so that a leaf node that has a value booked to it is changed to a parent node.

General information about hierarchies

An organization and generic type dimension can have one or more level-based hierarchies, or one or more parent-child hierarchies, but not both.

The following types of dimensions hierarchy features:

- In a classic model, the account dimension has only a one system-provided parent-child hierarchy but in the new model, multiple account hierarchies are allowed.
- For the date dimension, hierarchies are predefined based on the model granularity, and whether you have enabled fiscal time for the model. You can specify a default hierarchy in the settings for the date dimension. In the New Model, additional custom hierarchies may be added.
- The version dimension does not have a hierarchy, as different versions are separate and do not have parent-child relationships.

Flexible time hierarchies and date dimensions

Date dimension

The Date dimension defines the start and end dates of the model's timeline. It also specifies the granularity of the time units used in the model (year, quarter, month, week, or day.)

You can specify a default time hierarchy to display in stories, and optionally configure the date dimension to organize data by fiscal year instead of calendar year. Also, a model can contain more than one date dimension if needed.

The supported weekly time patterns are:

- 4-4-5
- 4-5-4
- 5-4-4
- 13x4

Date dimension modeling options

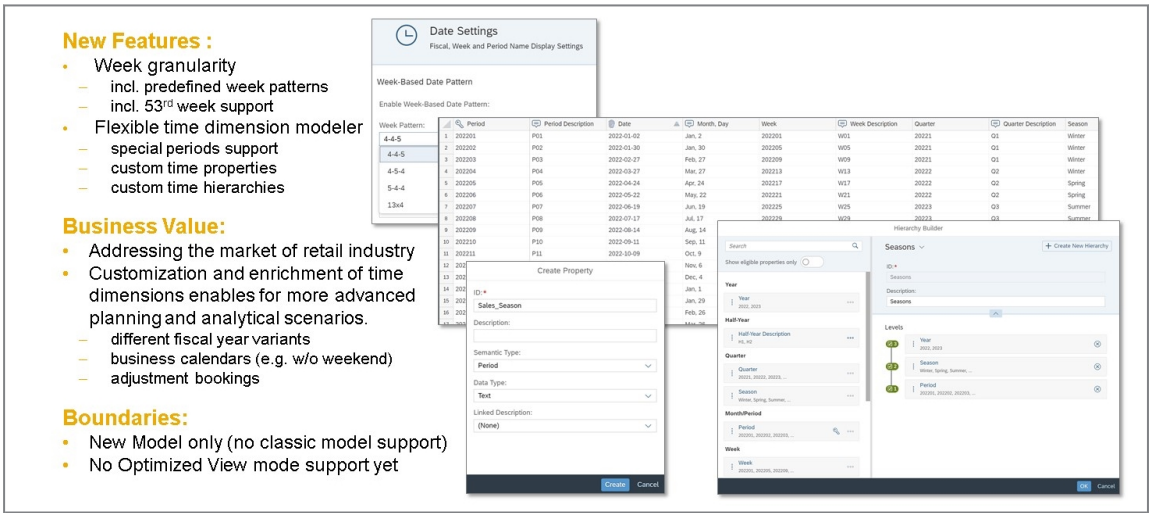


Figure 23: Date dimension modeling options

You can customize date dimensions members, hierarchies and properties to make time more flexible and suited to your planning and analytical needs. Flexible time hierarchies help you plan and analyze data along an individual time hierarchy. They also allow for company specific fiscal year calendars.

The concept of *special periods* refers to periods that are used to store adjustments postings. Some companies, for example, have a 12 month calendar along with 4 special periods ... one for each quarter.

Date dimensions can be maintained either manually or automatically by the system.

Leaving the management to the system means that the dimension's values are generated from the parameters within the Dimension Settings panel in the Modeler, and in the model preferences. Conversely, maintaining the date dimension manually allows you to edit the master data and manage all dimension members freely.

You can edit the dimension parameters just like you would with a standard date dimension. However, managing dimension members manually unlocks hierarchy management, the ability to create custom properties, and editing properties within the dimension view.

If you've set the date dimension to user-managed, you can edit the predefined hierarchies that come with the time granularity that are applied to the dimension.

You can add properties to date dimensions and specify dates of interest for your reporting or planning purposes, and account for specific periods like Holidays, or special sales seasons for instance.

Within a hierarchy, each property you create is assigned to a semantic type. A semantic type is a time unit assigned to a property within the hierarchy: Year, Half-Year, Quarter, Month, Period, Week, Day, and Other.

Note:
You can see an example of a weekly time pattern in the U00M_WeeklyHolidaySales model and U00S_SeasonalSales story in the SACM21 solution folder.

Custom date properties

You can add properties to date dimensions and specify dates of interest for your reporting or planning purposes and account for specific periods quickly, like national holidays or special sales periods, for instance.

Date hierarchies are level-based which means they are based on columns (properties) in the Data dimension. So if you need a custom hierarchy you need to add properties first. Each property you create is assigned to a semantic type. A semantic type is a time unit assigned to a property within the hierarchy: Year, Half-Year, Quarter, Month, Period, Week, Day, and Other.

The semantic type Other is reserved for custom properties that you want to attach to other properties as linked descriptions. In the example below, you can see the Month, Day, Week, and Month, Day properties used as descriptions and linked to their respective properties.

Easter is never the same week from year to year, and if you're working in sales, reporting on the Easter sales peak can be complex. Using customs properties, you can align the campaigns across the years using time series and make the comparison easier. Rather than trying to compare Easter sales with Easter being on different weeks, you can instead align the years with weeks leading up to Easter and the following weeks to have a single reference, and align the data points on a single time-line.

By flagging periods that are crucial to your business, you'll be able to pinpoint them promptly and speed up your analysis.

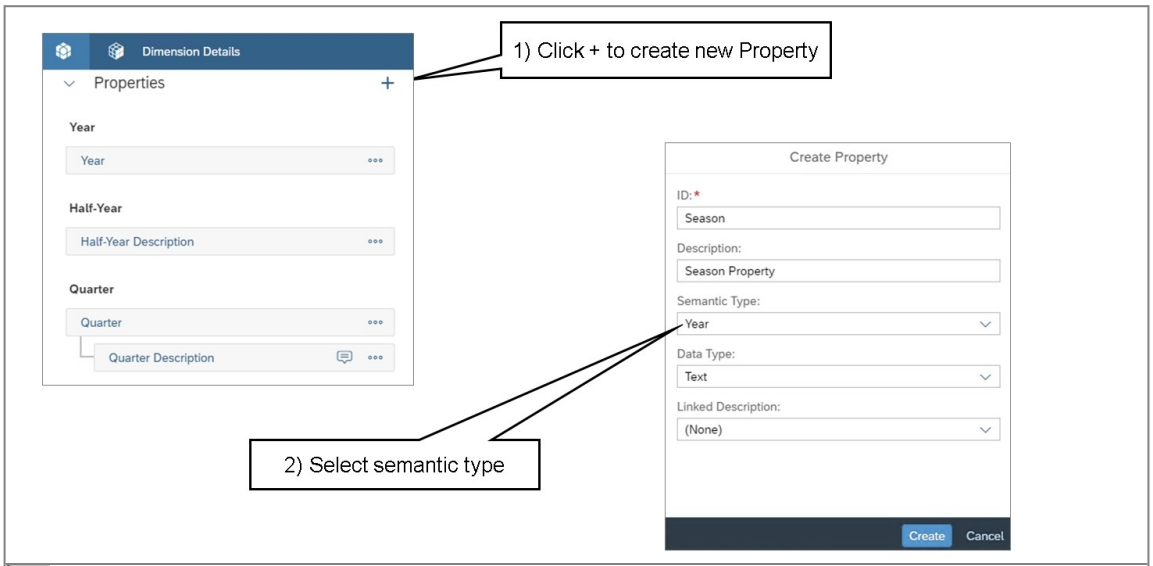


Figure 24: Create a custom date property

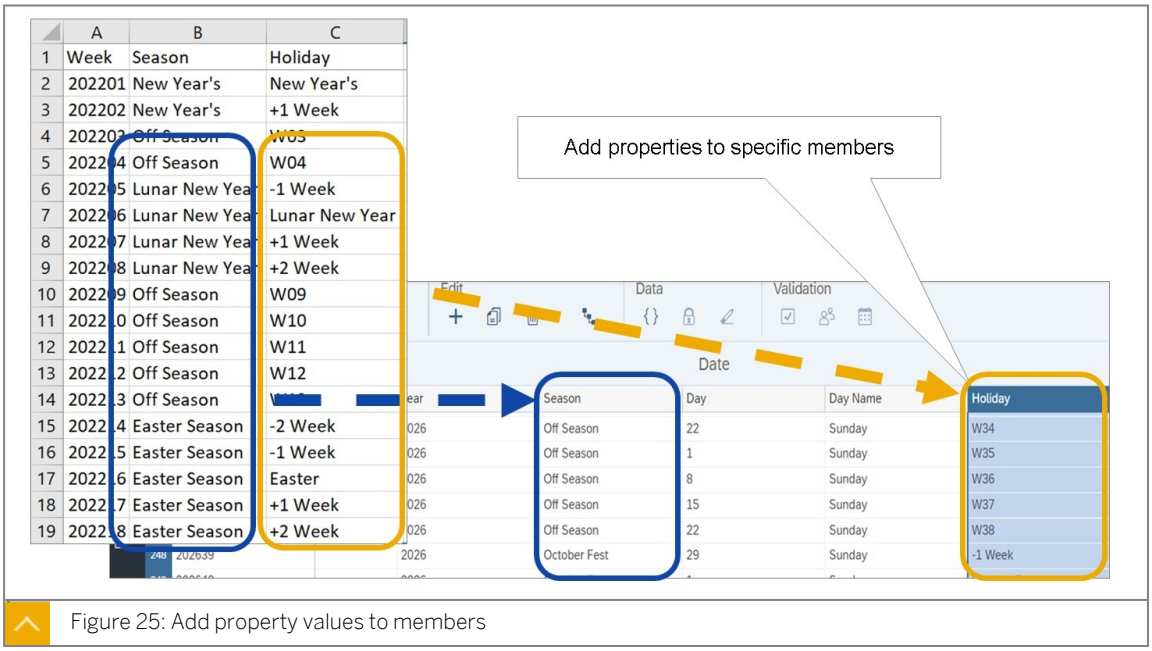


Figure 25: Add property values to members

The Season column is used to identify the seasons and will make up level 3 of the hierarchy. The Holiday column contains the assignment of each week to a season and will be the 2nd level of the hierarchy. The weeks will be in level 3.

In the *Add Property Values To Members* figure above, week 202216 is in middle of the Easter season. However, we are assuming for our example that the Easter celebration (buying patterns) spans over several weeks, including the previous two weeks. Therefore, week 202215 has a holiday value of -1Week, for example, 1 week prior to Easter week. Week 202214, has a holiday value of -2Week, for example, 2 weeks prior to Easter week, and so on.

Note:
The value -1Week or -2Week is arbitrary. The business requirement is to display the week before Easter week as *-1Week* when Easter season is drilled down on in the story.

Custom time hierarchies

With custom time hierarchies, you can customize date dimensions members, hierarchies, and properties to make time more flexible and suited to your planning and analytical needs. Flexible time hierarchies help you plan and analyze data using one or more hierarchies. They also allow for company specific fiscal year variants.

Date dimensions can be maintained both manually and automatically by the application. Leaving the management to application means that the dimensions values are generated from the parameters within the Dimension Settings panel in the Modeler, and in the model preferences. Conversely, maintaining the date dimension manually allows you to edit the master data and manage all dimension members freely.

You can edit the dimension parameters just like you would with a standard date dimension. However, managing dimension members manually unlocks hierarchy management, the ability to create custom properties, and editing properties within the dimension view.

If you've set the date dimension to user-managed, you can edit the predefined hierarchies that come with the time granularity that are applied to the dimension.

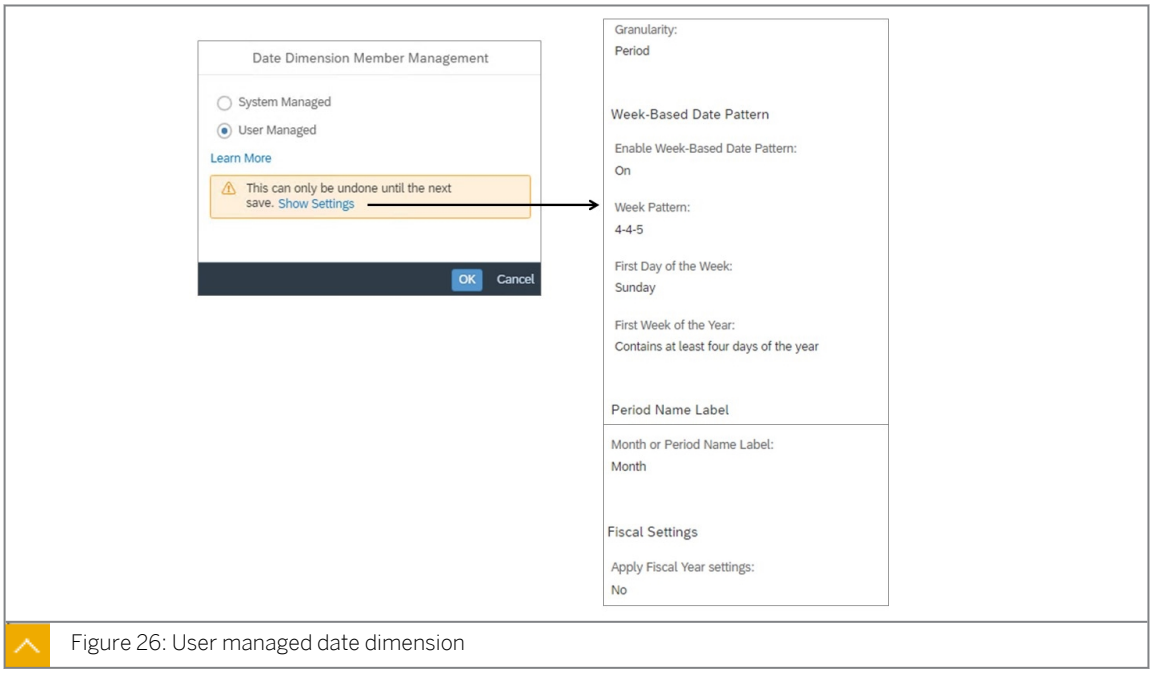


Figure 26: User managed date dimension

In the preceding *User Managed Date Dimension* figure, the date dimension is being set to User Managed. The 4-4-5 week pattern was set previously in the model preferences. Calendar year is being used since fiscal year was not turned on.

LESSON SUMMARY

You should now be able to:

- Explain the prerequisites required for creating hierarchies
- Create a dimension and hierarchy

Unit 2
Lesson 5

Importing and wrangling master data into a dimension

LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Import and wrangle data in a dimension

Import master data

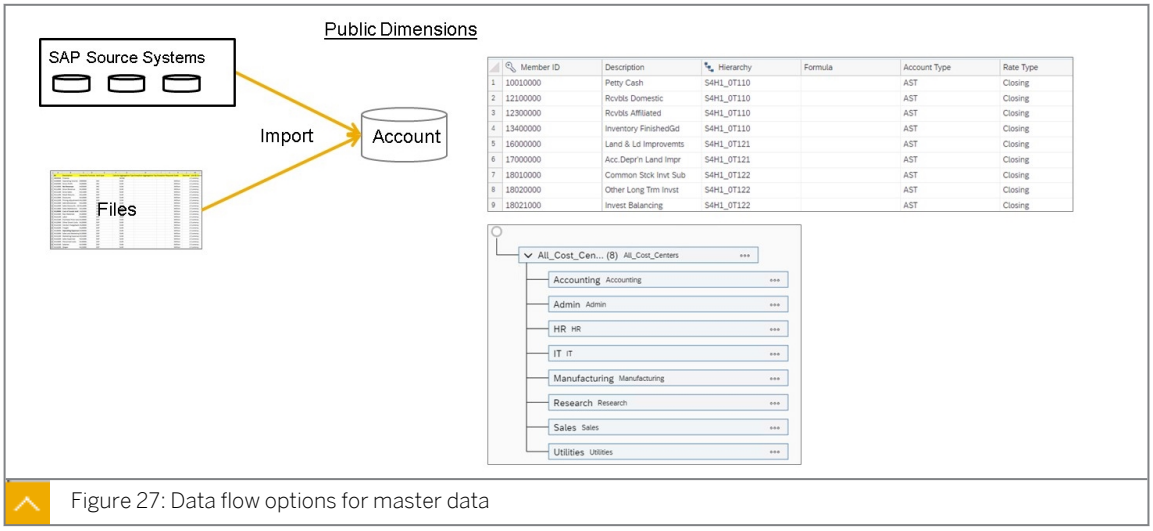


Figure 27: Data flow options for master data

Master data can be imported from SAP source systems or files. This includes the member id's, descriptions, hierarchies, and properties.

Data can be imported from non-SAP systems via custom connectors.

Table 3: Import Master Data - Workflow

File data source	SAP source system
Go to data management	Go to data management
Choose import data → File	Choose import data → Data source
Select the source file	Select the source system → select the connection
Transform data if needed	Choose the source system object
Map file columns to dimension columns	Select fields to be imported
Run the import → read the log	Add filters where needed

File data source	SAP source system
If records fail, debug and re-import	Transform data if needed
In order to run another import, repeat the process manually	Map file columns to dimension columns
	Run the import → read the log
	If records fail, debug and re-import
	In order to run another import, schedule the import query

Wrangle master data

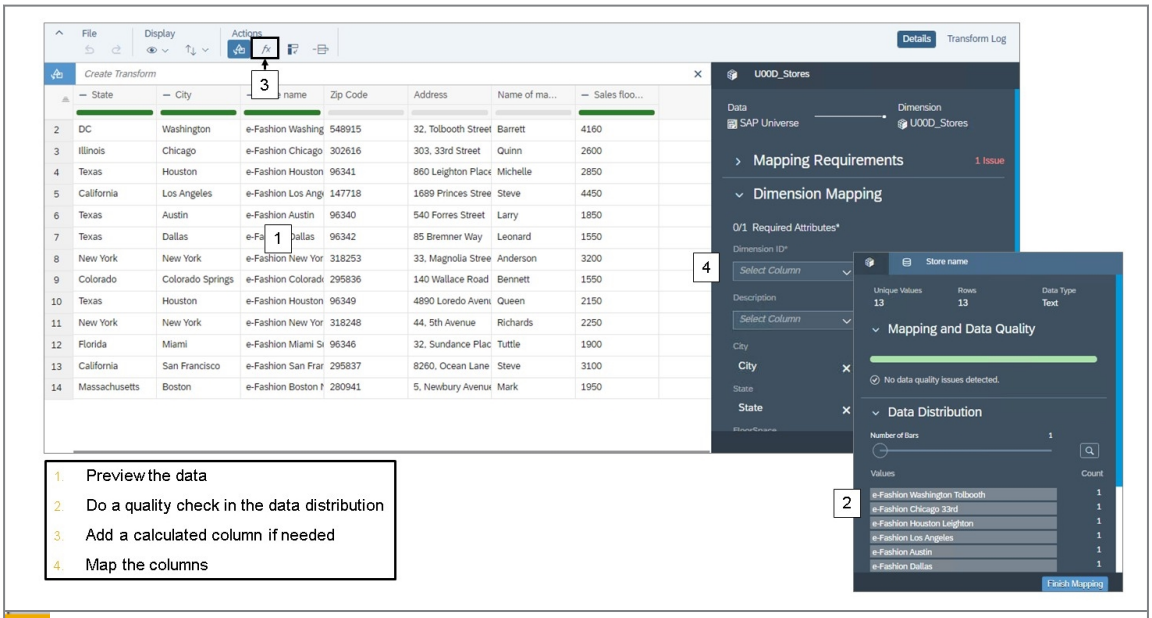


Figure 28: Data wrangling screen for master data

The data wangling screen allows the user to view the data and perform any necessary transformations.

Figure 29: Master data transformation examples

Concatenating columns is sometimes needed when dimension members are non-unique. For example, the same city name is used in multiple states so the state is inserted as a prefix. In SAP accounting, if there are multiple controlling areas, then it needs to be concatenated with cost center because the same cost center id is used in multiple controlling areas.

Smart Transformations lists suggested transformations to apply to the column, such as replacing the value in a cell with a suggested value. There are many transformation options, including but not limited to:

- Trim Whitespace - Remove spaces, including non-printing characters, from the start and end of strings.
- Duplicate Column - Create a copy of an existing column.
- Concatenate - Combine two or more columns into one. An optional value can be entered to separate the column values.
- Split - Split a text column on a chosen delimiter, starting from left to right. The number of splits can be chosen by the user.
- Change - Change a column to uppercase, lowercase, or title case.

LESSON SUMMARY

You should now be able to:

- Import and wrangle data in a dimension

Learning Assessment

1. Measure values are stored in models.

   Determine whether this statement is true or false.

   ☐ True

   ☐ False

2. Private dimensions can be updated during a model import.

   Determine whether this statement is true or false.

   ☐ True

   ☐ False

3. The account dimension has aggregation properties just like a measure.

   Determine whether this statement is true or false.

   ☐ True

   ☐ False

4. You can create public time dimensions.

   Determine whether this statement is true or false.

   ☐ True

   ☐ False

5. Person responsible can be enabled in a generic dimension.

   Determine whether this statement is true or false.

   ☐ True

   ☐ False

6. Level-based hierarchies use dimension properties as levels.

   Determine whether this statement is true or false.

   ☐ True

   ☐ False

7. Parent-child hierarchies are available in every dimension type.

   Determine whether this statement is true or false.

   ☐ True

   ☐ False

8. You can add calculated columns when importing into a dimension.

   Determine whether this statement is true or false.

   ☐ True

   ☐ False

Learning Assessment - Answers

1. Measure values are stored in models.

   Determine whether this statement is true or false.

   [X] True

   [ ] False

   Correct. Measure values are stored in models.

2. Private dimensions can be updated during a model import.

   Determine whether this statement is true or false.

   [X] True

   [ ] False

   Correct. Private dimensions can be updated during a model import.

3. The account dimension has aggregation properties just like a measure.

   Determine whether this statement is true or false.

   [X] True

   [ ] False

   Correct. The account dimension has aggregation properties just like a measure.

4. You can create public time dimensions.

   Determine whether this statement is true or false.

   [ ] True

   [X] False

   Correct. The system creates time dimensions. They are model-specific.

5. Person responsible can be enabled in a generic dimension.

   Determine whether this statement is true or false.

   [X] True

   [ ] False

   Correct. Person responsible can be enabled in a generic dimension.

6. Level-based hierarchies use dimension properties as levels.

   Determine whether this statement is true or false.

   [X] True

   [ ] False

   Correct. Level-based hierarchies use dimension properties as levels.

7. Parent-child hierarchies are available in every dimension type.

   Determine whether this statement is true or false.

   [ ] True

   [X] False

   Correct. Hierarchies are not available for the version dimension type.

8. You can add calculated columns when importing into a dimension.

   Determine whether this statement is true or false.

   [X] True

   [ ] False

   Correct. You can add calculated columns when importing into a dimension.

UNIT 3 Creating import models

UNIT OBJECTIVES

- Create an import model
- Explain data transformation functions in an import model
- Import data into a model
- Schedule data import
- Explain calculations created for import models
- Create calculated measures in a model
- Configure and translate currencies
- Explain the prerequisites required for creating hierarchies

Unit 3
Lesson 1

Creating an import model

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Create an import model

The basics of creating an import model

Creating import models

- Import models include both analytic & planning models
- There are three options when creating import models:
  - Create a blank model - this is the main workflow
    1. Create a blank model
    2. Add public dimensions
    3. Create private dimensions if needed
  - Create a model based on a file - files are used for ad-hoc / testing scenarios:
    1. Create model based on file
    2. Select the file
    3. Wrangle & map
    4. All dimensions are private in this case
  - Create a model based on a data source - Data sources are used for ad-hoc / testing scenarios:
    1. Create model based on data source
    2. Select import connection e.g. SAP Universe
    3. Select dimensions & measures
    4. Wrangle & map
    5. All dimensions are private in this case

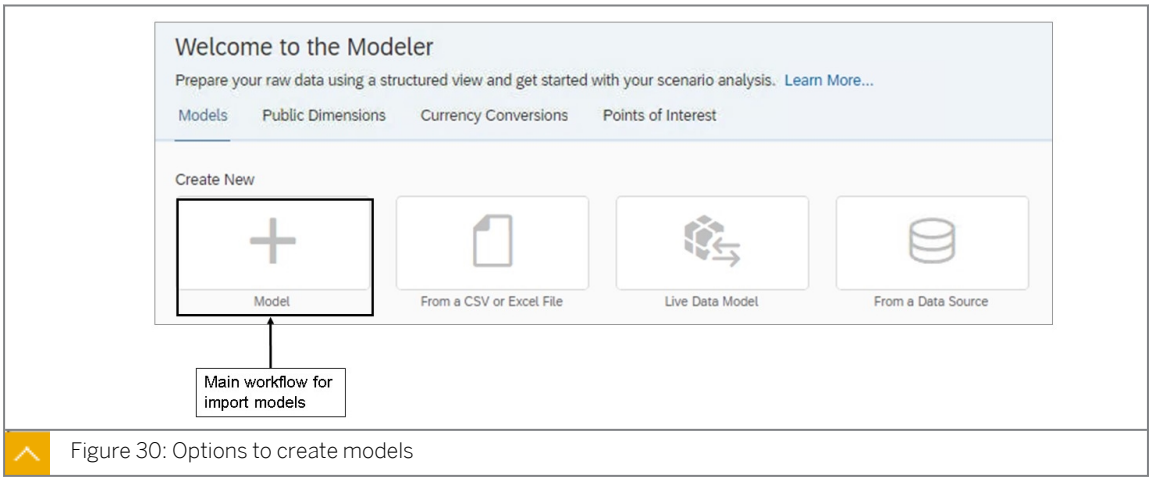


Figure 30: Options to create models

Note:
When creating models based on files and data sources, the classic wrangling screen is used and all of the dimensions are model-specific.

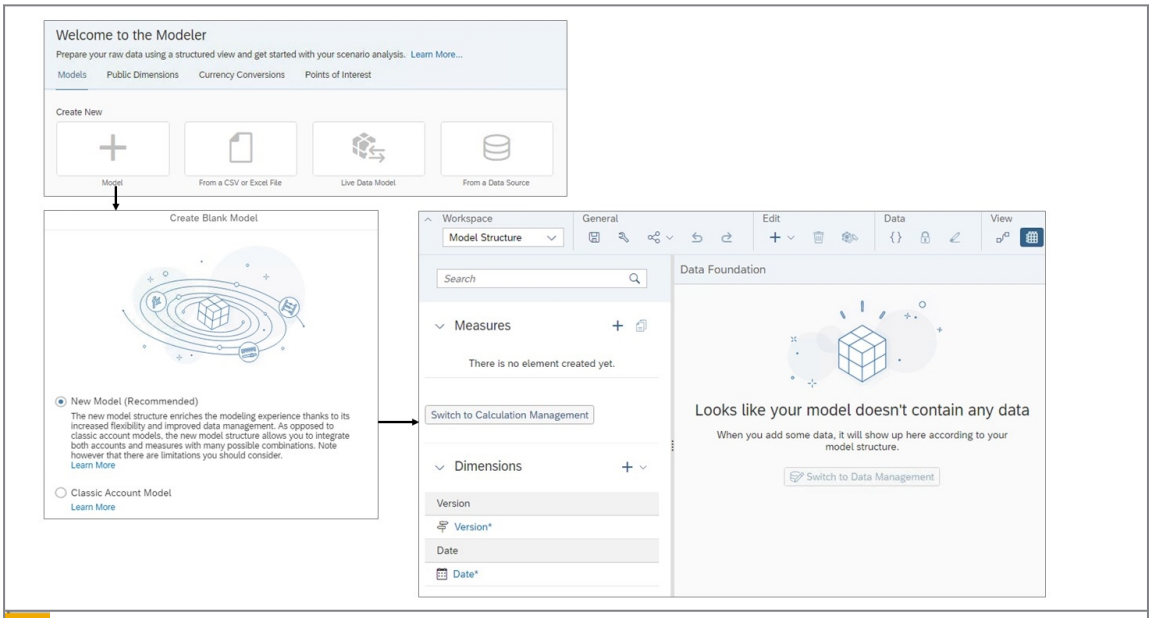


Figure 31: Main workflow to create an import model

Note:
When creating a new model, the default time range is from the current year to the current year + one. If the current year is 2024, the default time range will only allow data for 2024 and 2025.

Create import model workflow

Main steps for creating an import model:

1. Create public dimensions
2. Create the new type import model
3. Switch to analytic type or leave as planning type
4. Add existing public dimensions
5. Create at least one measure
6. Set model preferences if needed
7. Select the time range
8. Save the model
9. The model is ready for data imports

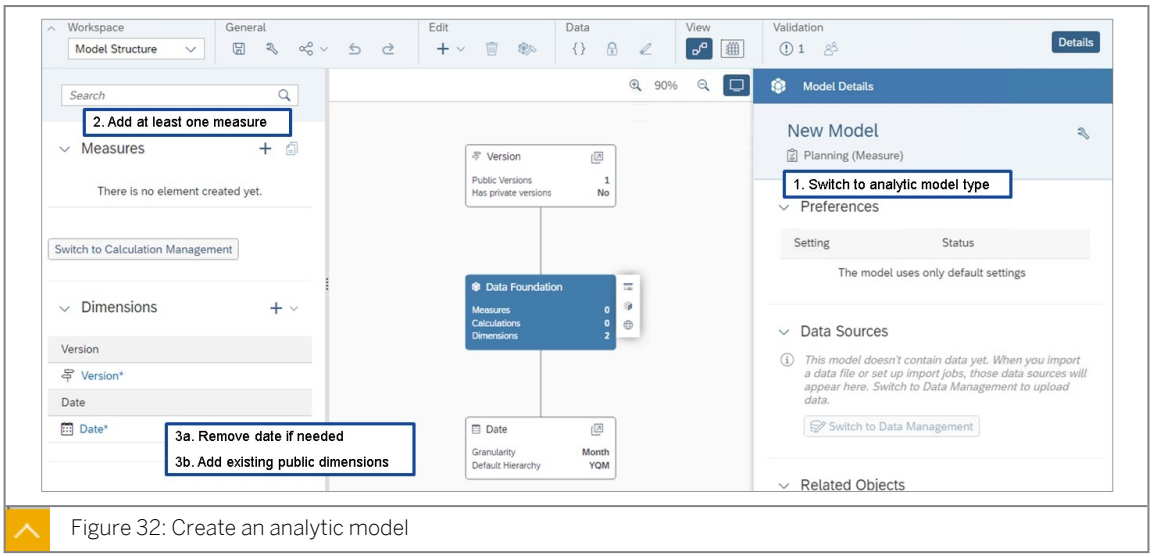


Figure 32: Create an analytic model

When you create a blank model, the default is the planning model type. That is why date and version are included by the system. If you change the model type to analytic, the version dimension goes away date becomes optional.

Note:
A typical model has a date dimension, account dimension, an organization dimension, several generic dimensions, and several measures.

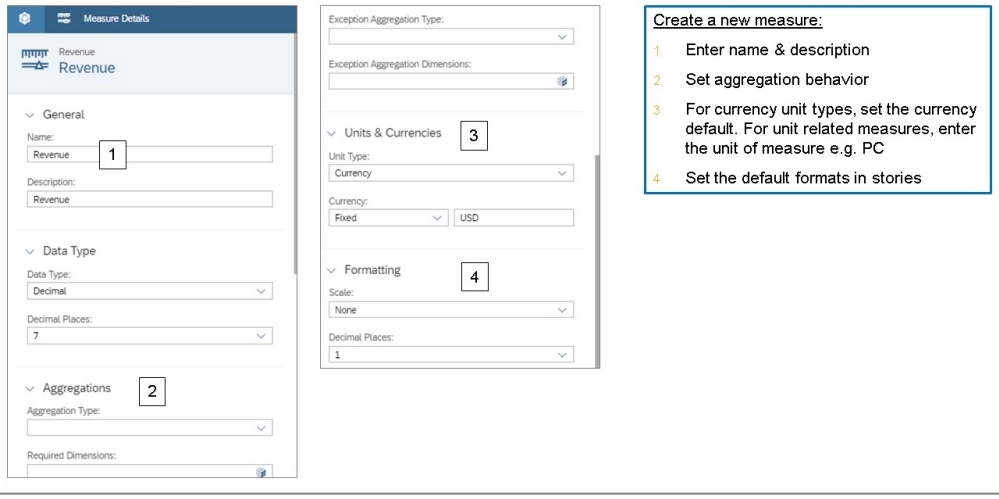


Figure 33: Add measures to a new model

When you add a measure to the model, the aggregation type will default to sum.

The exception aggregation type can be used to prevent inappropriate aggregation. For example, an FTE headcount measure could be set to an exception aggregation type of last and an exception aggregation dimension of date.

Currency related measures can either use a default currency or it can be derived from a dimension with the currency property.

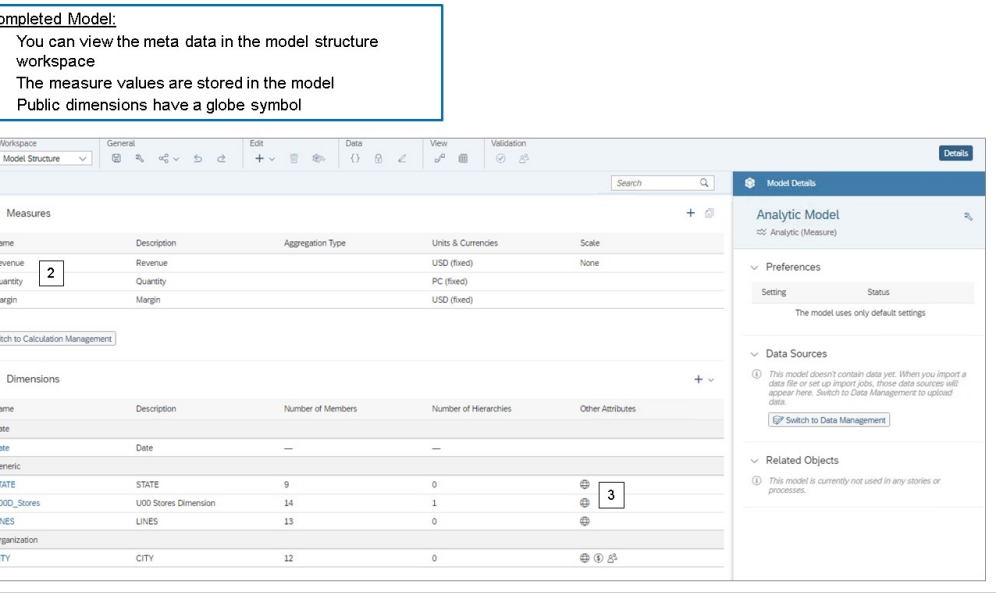


Figure 34: Example of a completed analytic model

Note:
Measure values are stored in the model however calculated and converted measures are refreshed on-the-fly and are not stored.

Preferences

Most preferences are made when the model is created and never changed.

Model preferences overview

- General settings: select the model type
- Language: request language translation
- Access and privacy: set data access and other controls
- Date settings: enable weekly based if needed, set date to calendar year or fiscal year
- Planning: set disaggregation behavior for planning models
- Currency: enable currency conversion
- Structure priority: set the tie breaker for data intersections for account and measure formulas
- Data and performance: settings to optimize performance for analytics and planning

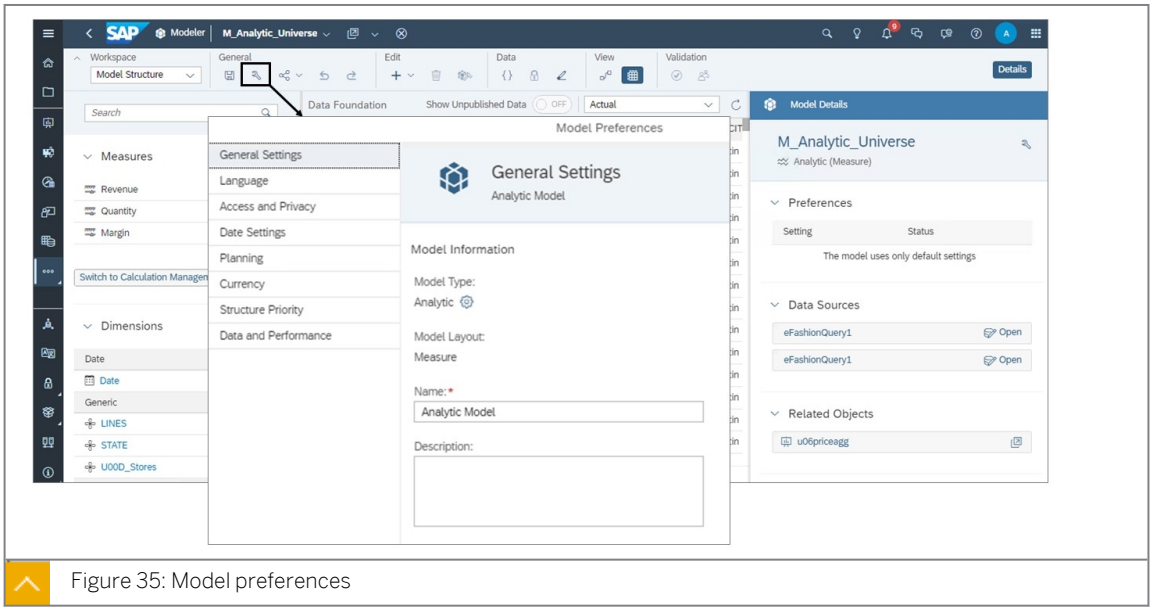


Figure 35: Model preferences

Viewing options

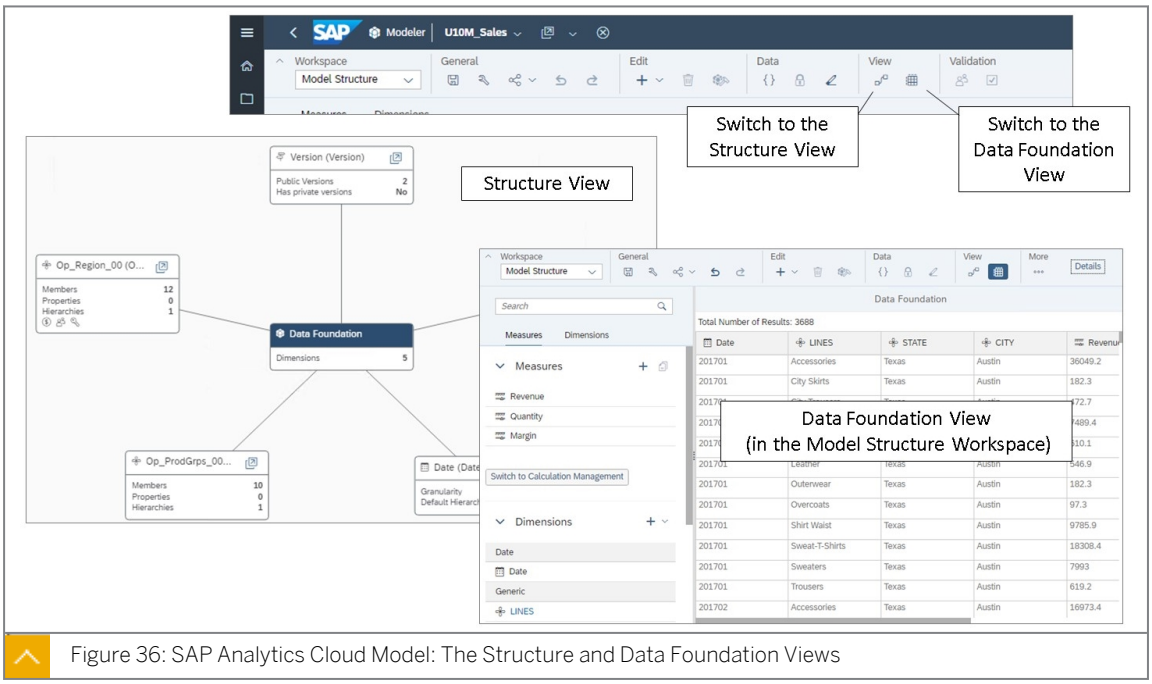


Figure 36: SAP Analytics Cloud Model: The Structure and Data Foundation Views

When creating a model, SAP Analytics Cloud has two viewing options:

- Structure View
- Data Foundation View

Structure View

The Structure view shows you a star schema diagram representing the contents of your model. This view helps you visualize how your fact data, attributes, and properties all relate to each other.

At a glance, you can see the model's dimensions surrounding the data foundation.

Additionally the dimension box shows you more information based on the dimension type. For example, the version dimensions shows how many public versions there are, and whether there are any private versions. For date dimensions, the information about granularity and default hierarchy, and (if added) whether the Fiscal Year setting has been applied.

If you want to add a new dimension or existing dimension to your model, you can do that from the toolbar, or from the Schema view.

Data Foundation View

The Data Foundation view shows you the fact table containing the raw, non-aggregated transactional data loaded into your model. The total number of results is the total number of rows of data, not including any filtering you have applied, across all versions.

In planning models, you can also switch between the different public versions of your data. For example, to see if data exists for a selected version.

The Structure and Data Foundation views work together with the dimension list and details panel to give you a consistent picture of your data. For example, if you select a dimension in the schema diagram, the other views all focus on that dimension.

LESSON SUMMARY

You should now be able to:

- Create an import model

Unit 3
Lesson 2

Importing and wrangling transactional data in a model

LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Explain data transformation functions in an import model
- Import data into a model

Import transaction data

Data can be imported into public dimensions and models from files as well as source systems.

> Note:
> The date and version dimension data is system-provided therefore there is no import option for date.

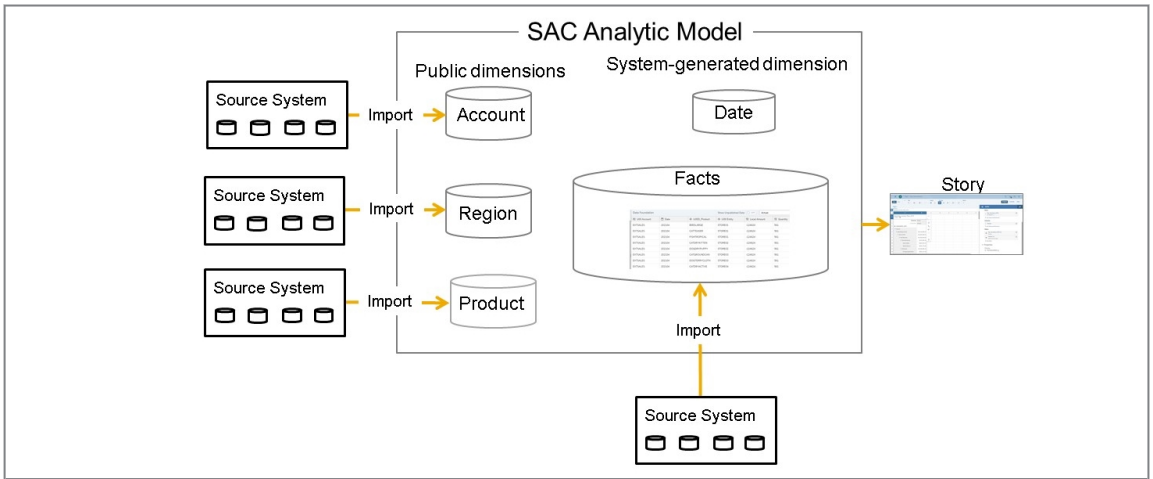


Figure 37: Import models data flow

Master data is normally imported before transaction data. When transaction data is imported, it is stored in the model's fact table.

A dimension can receive data from one source system or several depending on the scenario. The same is true for models.

Import transaction data workflow for files

Import transaction data from a file:

- Go to data management
- Choose import data → File
- Select the source file
- Set up import:
  - Prepare data
  - Map to target
  - Map properties
  - Review import
- Run the import → read the log
- If records fail, debug and re-import
- In order to run another import, repeat the process manually

Import transaction data workflow for SAP Source Systems

Import transaction data from an SAP Source System:

- Go to data management
- Choose import data → Data source
- Select the source system → select the connection
- Choose the source system object
- Select fields to be imported
- Add filters where needed
- Set up import:
  - Prepare data
  - Map to target
  - Map properties
  - Review import
- Run the import → read the log
- If records fail, debug and re-import
- In order to run another import, repeat the process manually or schedule it

The worklow for importing transaction data is similar to master data. One key difference is the *set up import* activities. The *set up import* process is made up of four work steps as shown above.

Note:
The workflows above describe the import process for new models not classic models.

When you import data into a model from an SAP source system, you will need to select the fields to import. In addition, you can set a filter for the import. This may include a filter for year and period. After creating the import query, the filter can be changed but the field selection can not be changed.

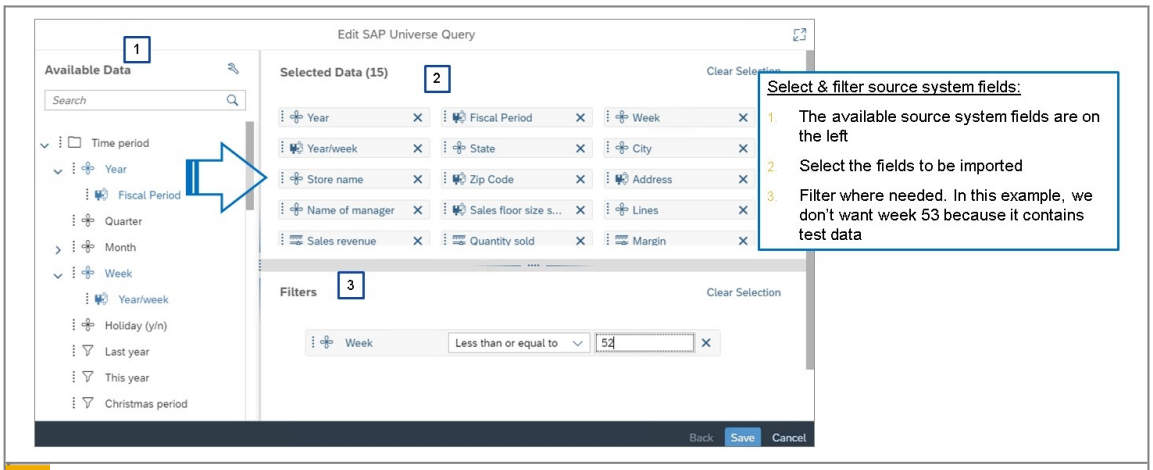


Figure 38: Selecting fields to import

When the import is created, it appears in the data management workspace of the model with a hyperlink: *Set Up Import*.

Note:
When creating the import query, the field selections cannot be edited after they are created. The filter however, can be edited.

Wrangle transaction data

Wrangle data

Now that you have a feel for the potential data targets such as dimensions and models, let's head towards the import process.

Flat files can be imported to both public dimensions as well as models, however, the main source of master and transaction data for most customers will be from SAP systems such as SAP Business Warehouse and SAP S/4HANA.

When importing from SAP systems, the imports can be scheduled. As a matter of fact, dimensions can be imported in a sequence that includes the model import.

Prepare the data

As previously discussed in this course, data used in SAP Analytics Cloud models for stories and analytical applications may need to be modified for reporting and analysis purposes. For example, data from a source system that is imported into a model may have multiple columns for employee names - first name and last name. But for stories and analytic applications, designers may need only one column that combines both first and last names.

In the prepare data step, you can perform easy wrangling, do quality checks, perform complex transformations, and change the job settings.

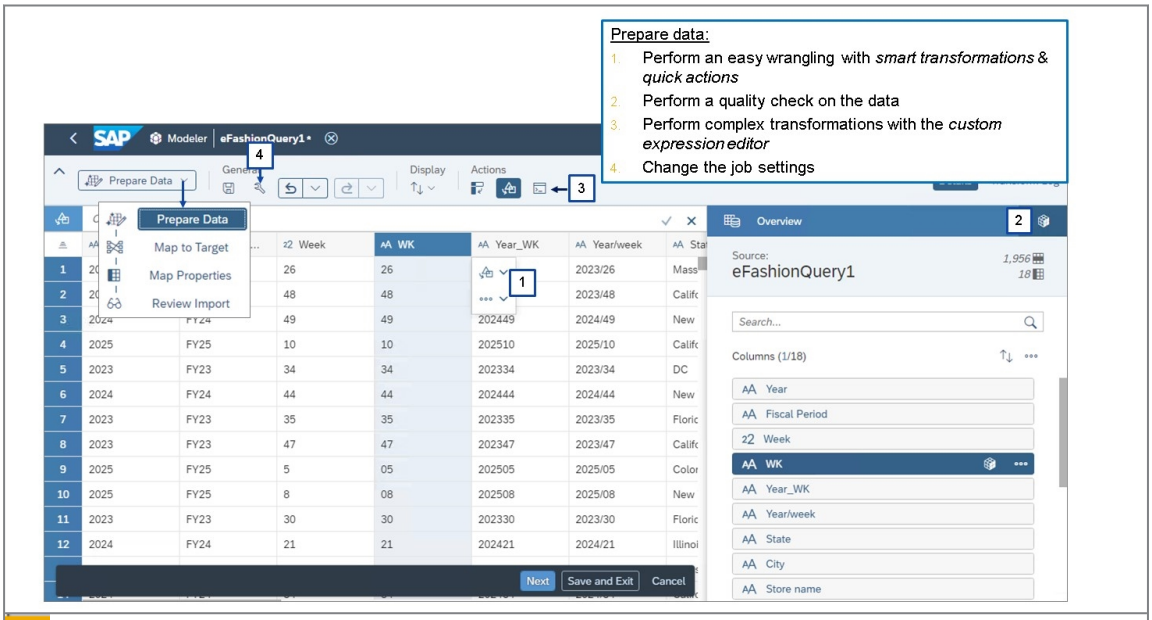


Figure 39: The prepare data step

Note:
For source system imports, the four steps in the import workflow can be edited after it is run.

Custom expression editor

When importing into a new type model, simple transformations are available via the smart transformation and quick actions. If more complex transformations are needed, the custom expression editor is available.

The custom expression editor features includes:

- Ability to create formulas to perform more complicated transformations
- Available for new model and dataset imports
- Auto-complete
- In-line help

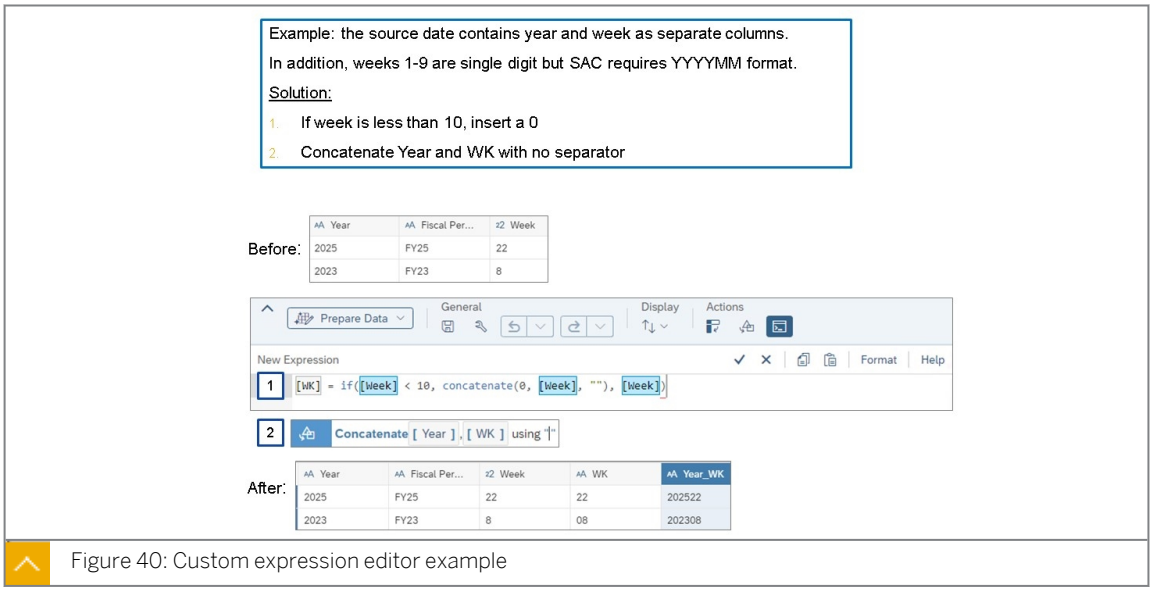


Figure 40: Custom expression editor example

The custom expression editor is the more recent option as compared to the “add calculated column” option for master data imports.

Job settings

During the import process, the job settings can be maintained:

- **Import method:** These selections are important when importing data again for the same data region
- **Reverse sign by account type:** If the source system stores revenue, equity, and liabilities with positive values, then they can be reversed based on the account type property in the account dimension.
- **Update local dimensions with new members:** If the model contains a private dimension, then its members can be updated with new members during the model import
- **Conditional validation:** This is used to validate members to make sure they are not parent members in a hierarchy

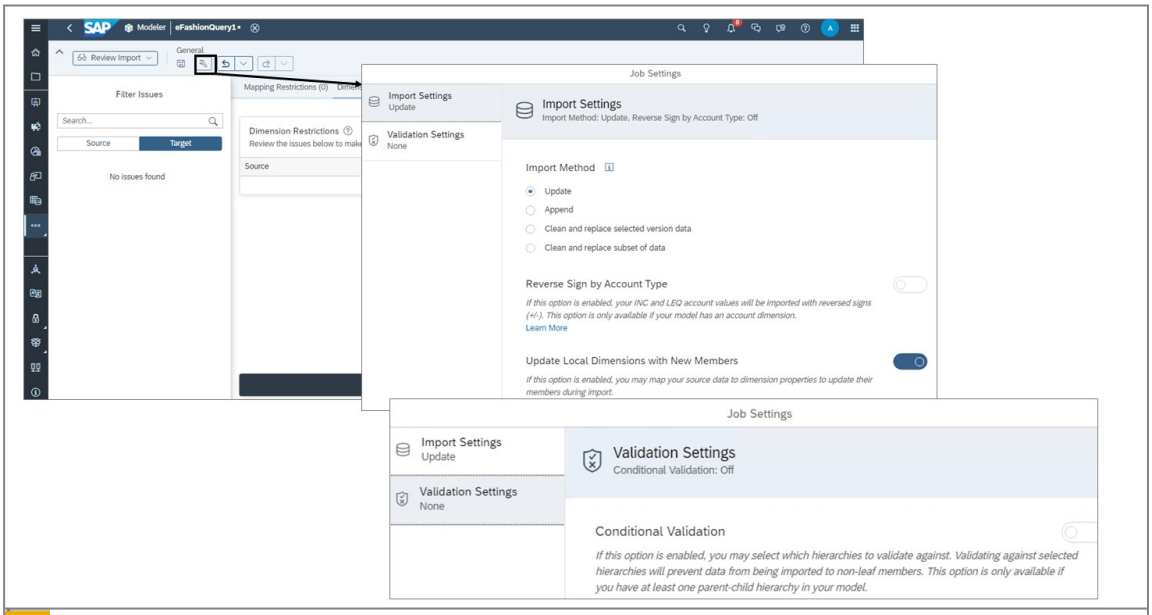


Figure 41: Job Settings

Map to target

In the **Map to target** step, the system will map fields automatically where possible. If the source and target fields names are different, then the mapping can be completed using drag and drop. When importing from an SAP source system, the mapping can be edited, if needed.

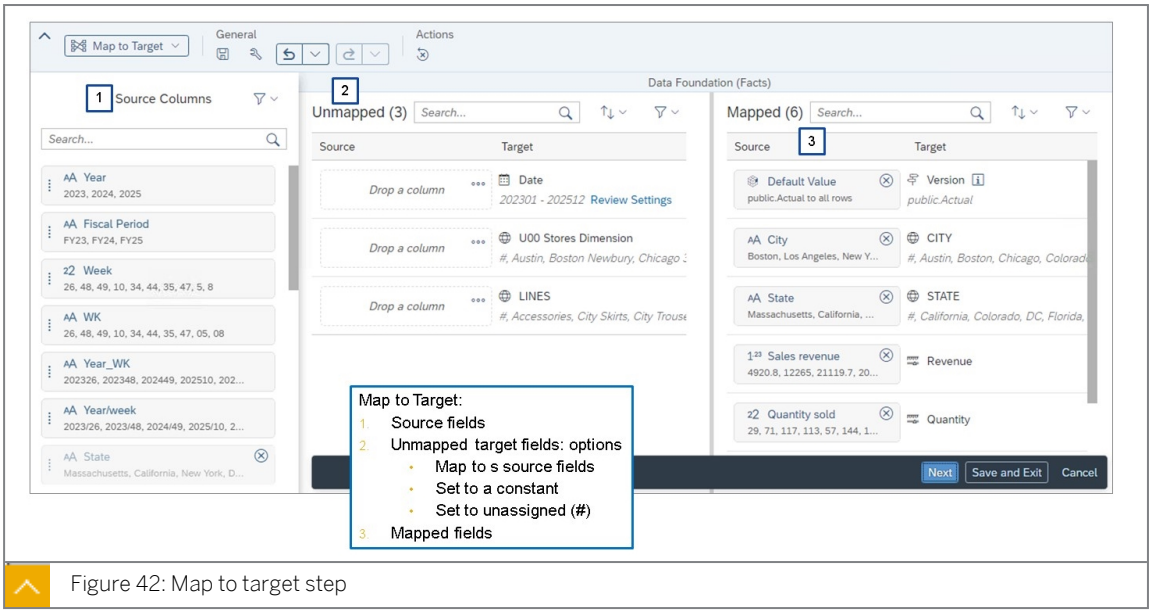


Figure 42: Map to target step

Map properties

The **Map properties**step can be used to map properties of private dimensions if there are any. The only way to import into a private dimension is during the model import. Also, when importing into an analytic model, the Map Property step indicates that all records have been assigned to the public actual version.

Note:
All target fields must either be mapped or use a default value or unassigned. Some source fields may be unmapped if they are not needed.

Review Import

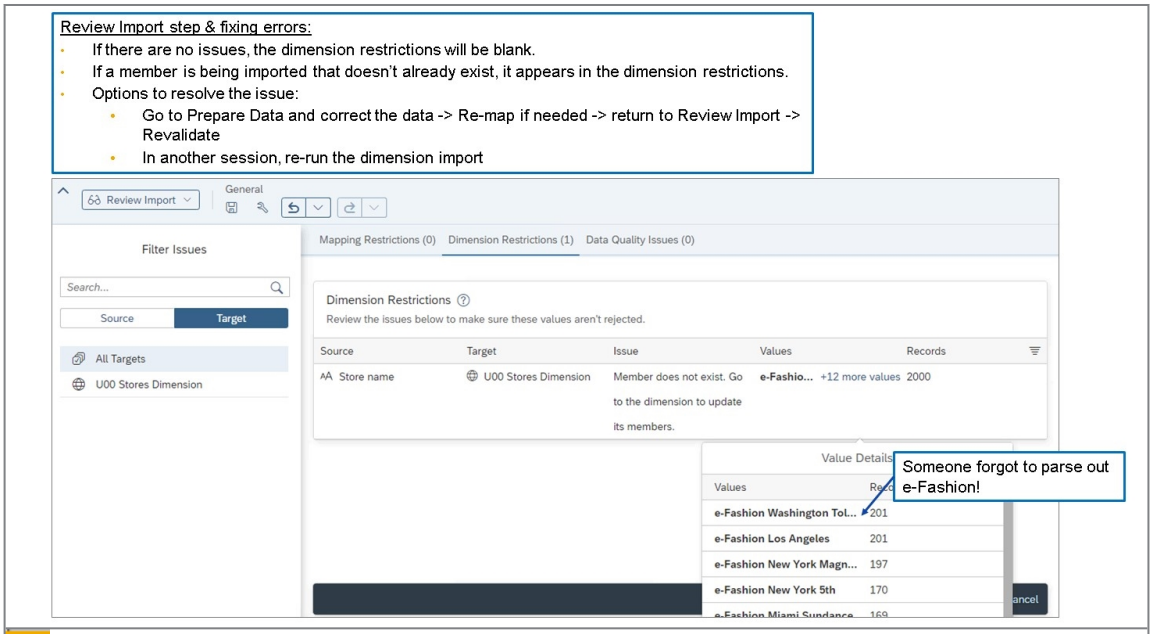


Figure 43: Review import step

Another issue that may arise is when member IDs are in the import source but not in the target dimension. This should not happen very often since master data is updated before importing into the model. The solution would be to add the new dimension member ID and re-validate the review import step (and then run if OK).

Import query options

After the import process is complete, there are several important options in the data management workspace:

- **Edit** this allows to change the filter values in the import query
- **Schedule** is where you can set up the import query to run periodically (e.g. daily).
- **Select import method** is used to switch the import method (e.g. from update to clean and replace).
- **Refresh** is used to run the import again

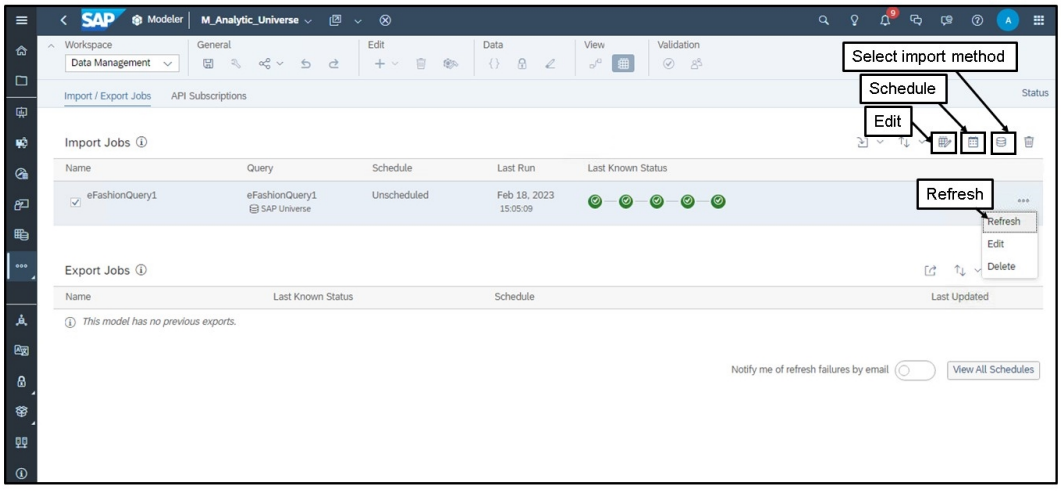


Figure 44: Import query options

Combine data from two sources

Combine data from two sources

Another option for wrangling data includes combining data from two sources.

The figure *Wrangling - Combine Data from Two Sources* is an example of this concept.

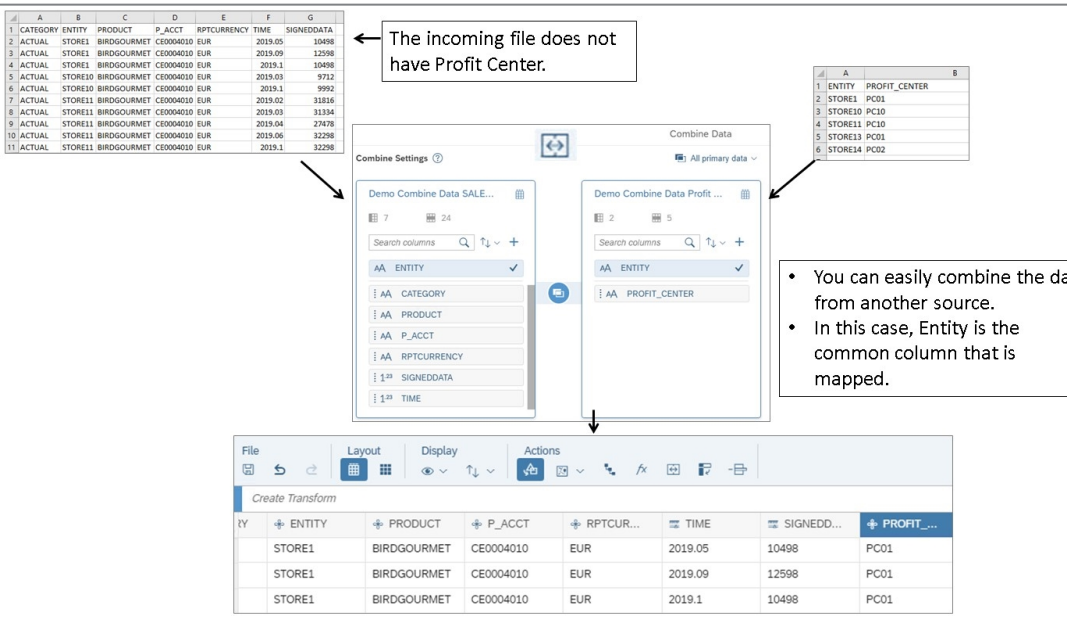


Figure 45: Wrangling - combine data from two sources

When importing into a classic model, you can merge data from multiple sources. The prerequisite is that you need some common members in the source data.

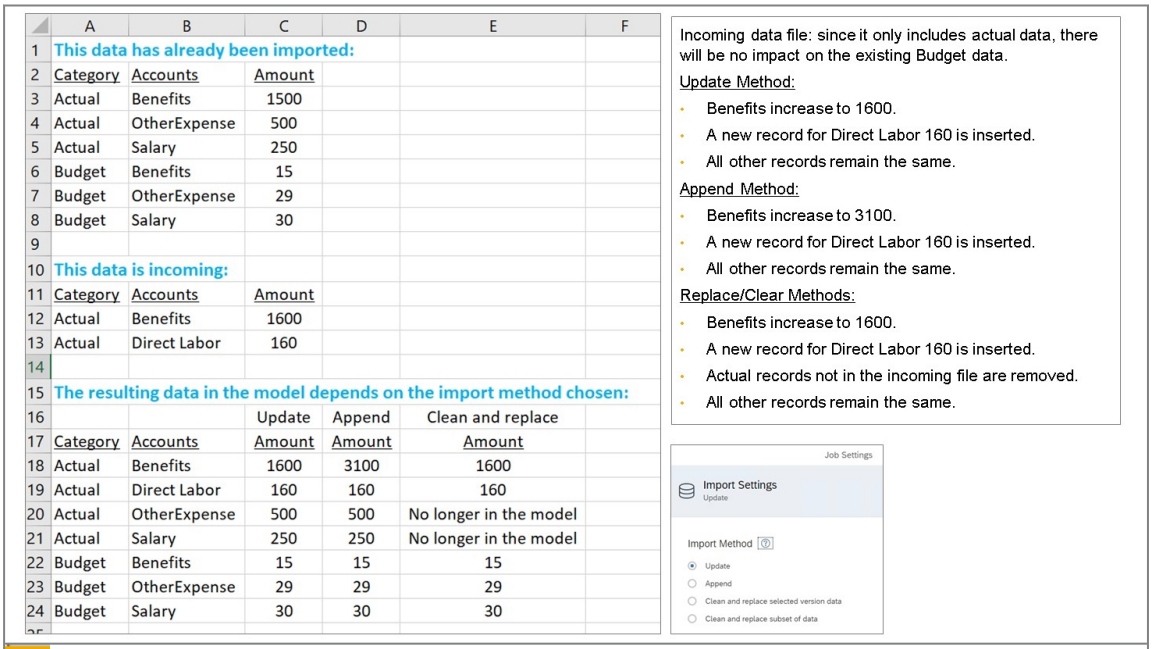
Update methods

Quite often, monthly data is imported every day into SAP Analytics Cloud models. In cases like this, the designer can use the update options to prevent data duplication. Also, when importing from SAP systems, the import is typically filtered to the current period.

When data is imported multiple times for the same data region, the update method is used to control the data update.

A choice of four import methods are available:

Import Method	Description
Update	Updates the existing data and adds new entries to the target model. The scope of this update is based on a combination of all dimensions in the target model. For a more refined scope, use either the *Clean and replace selected version data* or *Clean and replace subset of data update* options.
Append	When using append there are two options: 1. Reject duplicated records in the model. 2. Aggregate duplicated rows in the model.
Clean and replace selected version data	Deletes the existing data and adds new entries to the target model, only for the versions that you specify in the import. You can choose to use either the existing version or specify a new version under *Version*. For example, if you specify to import data for the *Actual* version, only the data in the *Actual* version is cleaned and replaced. Other versions, for example *Planning*, are not affected.
Clean and replace subset of data	Replaces existing data and adds new entries to the target model for a defined subset of the data based on a scope of selected versions using either the *Existing Version* or *New Version* buttons. You can also limit the scope to specific dimensions. To define a scope based on a combination of dimensions, select + *Add Scope* and use the *Select a dimension* field to specify a dimension. When a *Date* dimension is defined in the scope, the time range in the source data (calculated based on the target model's granularity) combined with other dimensions in the scope will determine what existing data is cleaned and replaced. If, for example, *Date* and *Region* dimensions are defined as part of a scope, only entries that fall within the time range and match *Region* from the source data will be replaced in the target model. Existing data that does not match the scope will be kept as is.



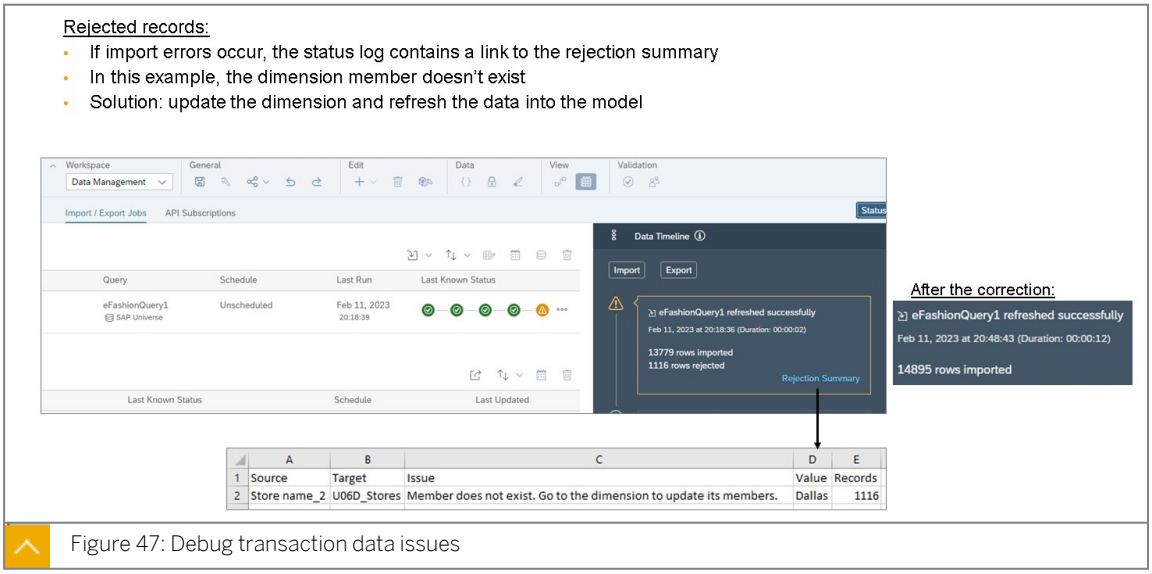
	A	B	C	D	E	F
1	This data has already been imported:					
2	Category	Accounts	Amount			
3	Actual	Benefits	1500			
4	Actual	OtherExpense	500			
5	Actual	Salary	250			
6	Budget	Benefits	15			
7	Budget	OtherExpense	29			
8	Budget	Salary	30			
9						
10	This data is incoming:					
11	Category	Accounts	Amount			
12	Actual	Benefits	1600			
13	Actual	Direct Labor	160			
14						
15	The resulting data in the model depends on the import method chosen:					
16			Update	Append	Clean and replace	
17	Category	Accounts	Amount	Amount	Amount	
18	Actual	Benefits	1600	3100	1600	
19	Actual	Direct Labor	160	160	160	
20	Actual	OtherExpense	500	500	No longer in the model	
21	Actual	Salary	250	250	No longer in the model	
22	Budget	Benefits	15	15	15	
23	Budget	OtherExpense	29	29	29	
24	Budget	Salary	30	30	30	

Figure 46: Import method comparison

Debug transaction data

When importing data into a model, the review import step will flag records with issues. If the import is run while there are remaining issues, a rejection summary will appear in the data management workspace. In addition, the log will indicate the number of accepted vs. rejected records.

> Note:
> If there is an issue with a scheduled import, the schedule status log can be access via *Connections* in the navigation bar.



	A	B	C	D	E
1	Source	Target	Issue	Value	Records
2	Store name_2	U06D_Stores	Member does not exist. Go to the dimension to update its members.	Dallas	1116

Figure 47: Debug transaction data issues

In the *Debug transaction data issues* figure above, the missing member was updated into the dimension. In the model's data management workspace the import query was refreshed.

When some incoming records are okay but some have issues, the good records are imported and the bad records are identified in the rejection summary file.

LESSON SUMMARY

You should now be able to:

- Explain data transformation functions in an import model
- Import data into a model

Scheduling data imports

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Schedule data import

Schedule data imports

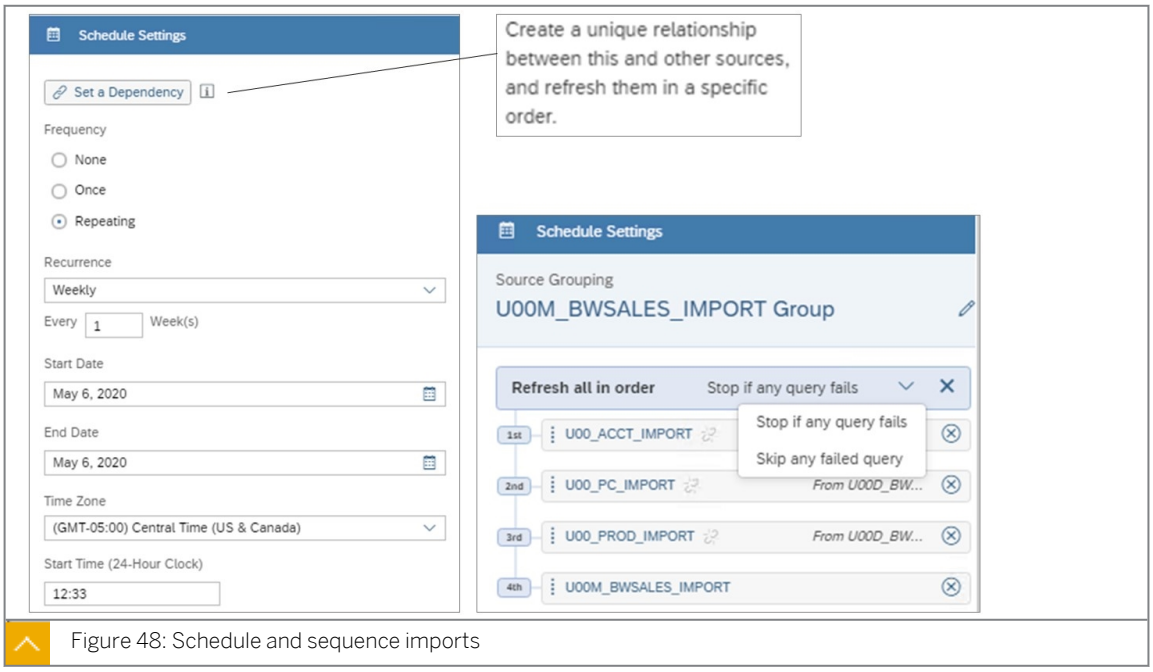


Figure 48: Schedule and sequence imports

You can import data from multiple queries (and data sources) into a model, and each of these imports can be separately scheduled.

If you want to run multiple Import data jobs together, in a specified order, create a source grouping. A grouping can include jobs from public dimensions as well as the model. Running the grouping refreshes the public dimensions and model together.

Scheduled jobs can also be viewed, deleted, and refreshed via: *Navigation Bar → Connections → Schedule Status.*

If any rows in the dataset were rejected, you can select *Download rejected rows* to save the rejected rows as a .csv file. You can then examine this file to see which data was rejected. You could fix the data in the .csv file and then import the .csv file into the model using the *Import Data* workflow, or fix the data in the source system.

LESSON SUMMARY

You should now be able to:

- Schedule data import

Creating formulas

LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Explain calculations created for import models
- Create calculated measures in a model

Dimension formulas

Dimension member formulas

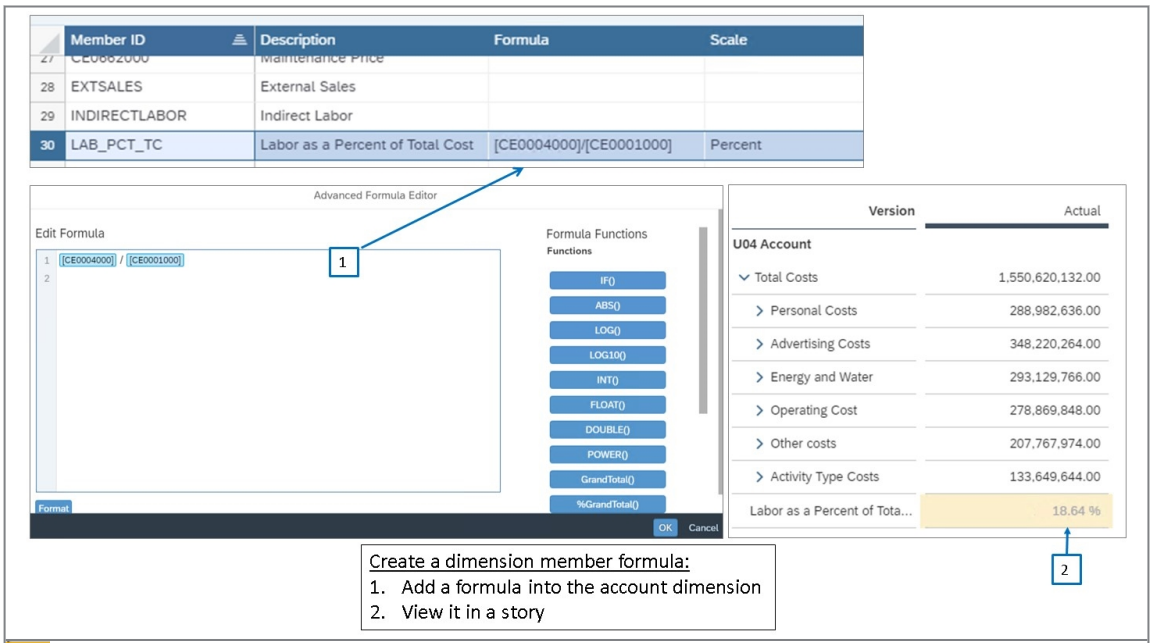


Figure 49: Dimension formula example - calculate labor as a % of total cost

Dimension member formulas are configured in account dimensions to perform on-the-fly calculations such as gross margin % or labor as a percent of total operating cost. Here are some other key features of dimension member formulas:

- You can use dimension member formulas to calculate account values that are restricted to other dimension member values such as prior year revenue.
- A dimension formula is available any story that uses a model that contains the account dimension with a formula.

- An account dimension formula can be used as-is (from the dimension) or they can be modified in specific models. In this scenario, the model can contain its own version of the dimension formula.
- You can also use dimension member formulas to calculate account values that are restricted to other dimension member values such as prior year revenue.
- Dimension member formulas can be viewed in stories by displayed the formula bar.

Calculated measures

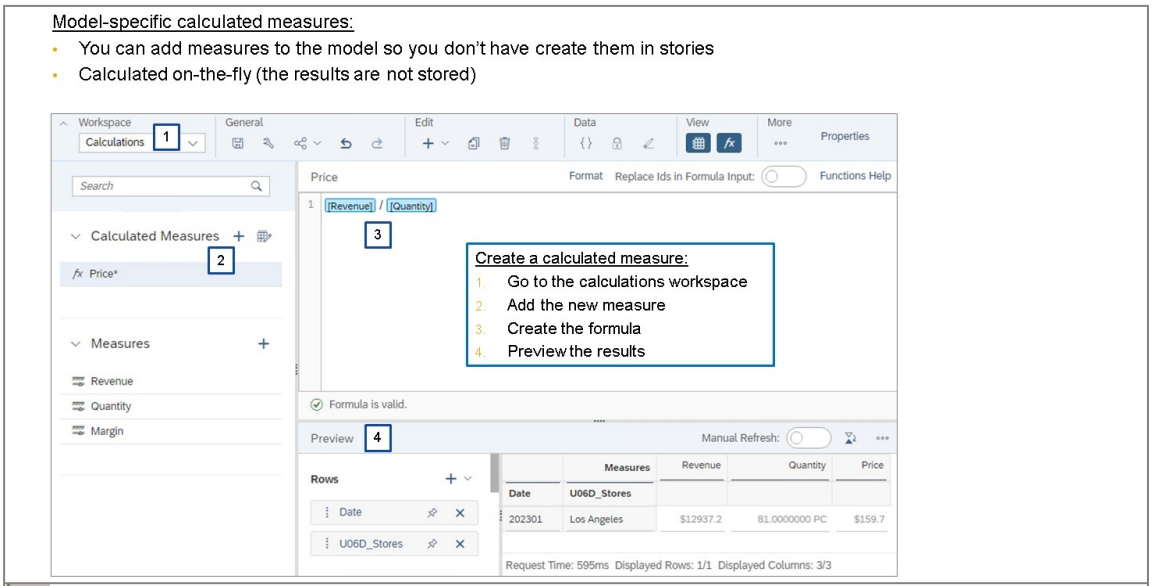


Figure 50: Model-based calculated measures

Calculated measures do not appear in the data foundation view because the results are not stored.

Calculated measure aggregation settings

Calculated measure aggregation settings prevent unwanted summation. For example, the default aggregation (sum) is fine for a calculated measure like: Revenue = Price x Quantity.

However, if the calculated measure is: Price = Revenue / Quantity, then summation is not valid. To avoid summation, the exception aggregation type is set to None and the exception aggregation dimensions is set to date and store.

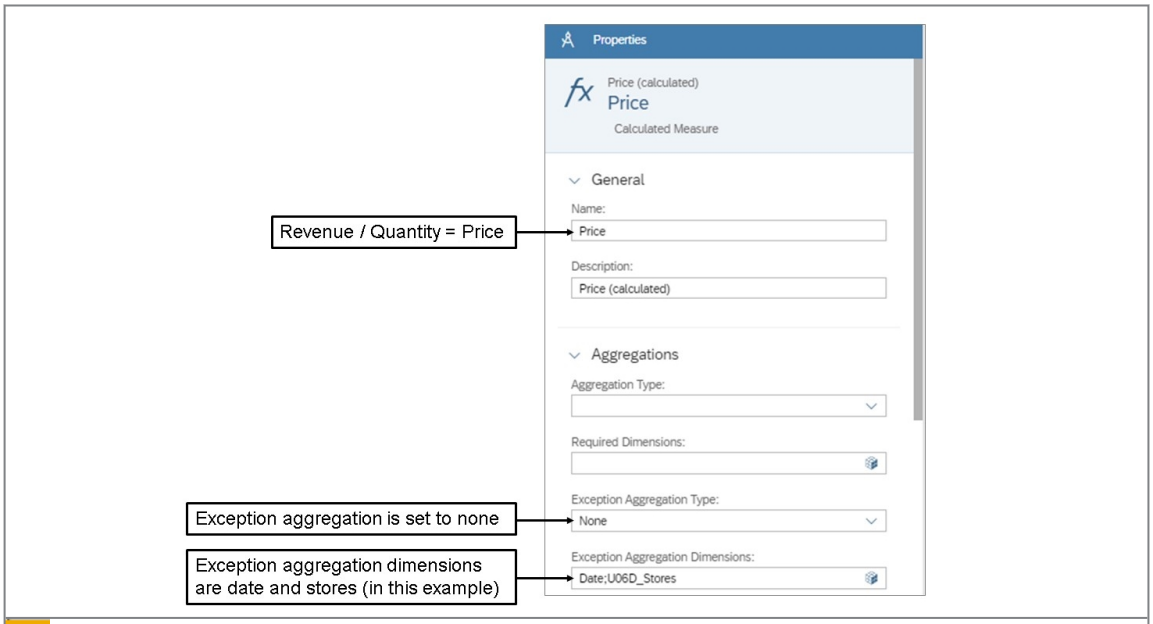


Figure 51: Calculated measure aggregation

LESSON SUMMARY

You should now be able to:

- Explain calculations created for import models
- Create calculated measures in a model

Unit 3
Lesson 5

Translating currency

LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Configure and translate currencies

Currency translation overview

SAP Analytics Cloud analytic models can easily handle most currency scenarios without a separate currency dimension by using the model and story currency conversion measures.

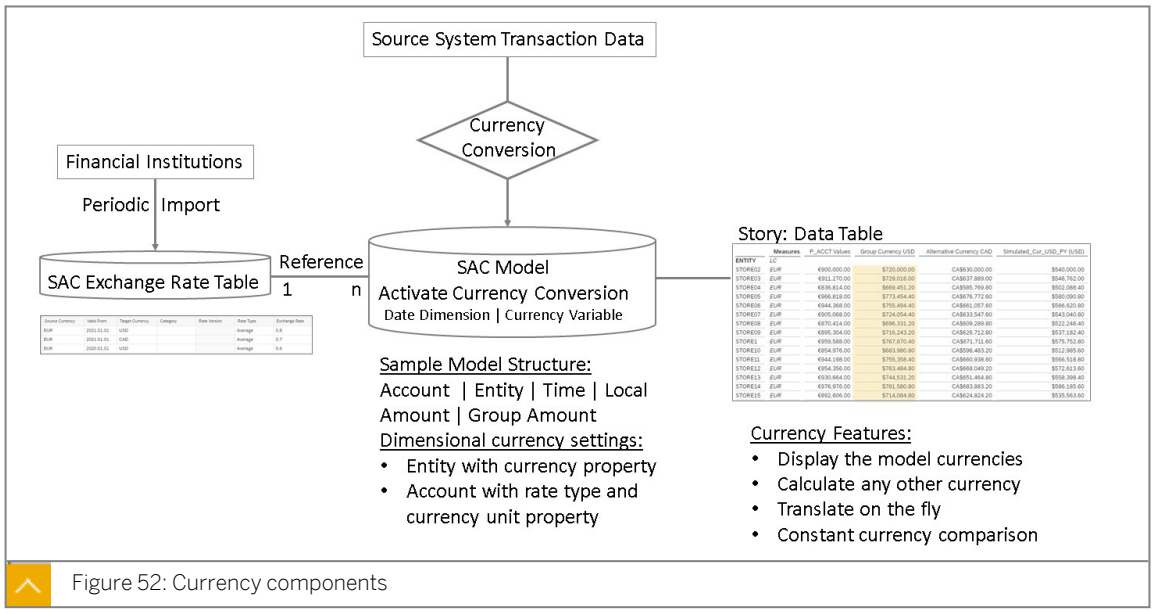


Figure 52: Currency components

Note:
New Models users can enable a currency variable in the model preferences to set a default target currency for conversions measures.

Note:
Currency tables can be updated via:

- Flat files
- BPC rate models
- BW Queries

SAP Analytics Cloud currency translation - key points

- Exchange rates are stored in the SAP Analytics Cloud currency translation table.
- Exchange rates originate from financial institutions and are automatically or manually updated into SAC.
- Currency use case examples include:
  - Display corporate (group) and local operating currencies in stories.
  - Import and plan in local currency and immediately translate to group currency.
  - Translate with category-specific exchange rates.
  - Import both local and group currency actual values.
  - Analyze trends in constant currency.

Note:

- The term **Local Currency** (LC) refers to the functional or operating currency of a company, for example. A company in the US has an LC of USD, and so on.
- The term **Group Currency** refers to the corporate level currency. A company in the US that has a German parent therefore has a group currency of EUR, for example.
- The term **Translation** is being used interchangeably with **conversion**.
- The concepts in this lesson concern planning concepts only, not month-end closing currency translation.

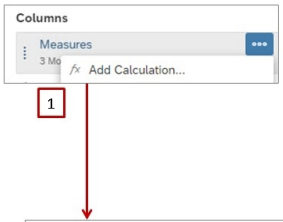
ENTITY	Measures	P_ACCT Values	Group Currency USD	Alternative Currency CAD	Simulated_Cur_USD_PY (USD)
	LC	2	3	4	5
STORE02	EUR 1	€900,000.00	$720,000.00	CA$630,000.00	$540,000.00
STORE03	EUR	€911,270.00	$729,016.00	CA$637,889.00	$546,762.00
STORE04	EUR	€836,814.00	$669,451.20	CA$585,769.80	$502,088.40
STORE05	EUR	€966,818.00	$773,454.40	CA$676,772.60	$580,090.80
STORE06	EUR	€944,368.00	$755,494.40	CA$661,057.60	$566,620.80
STORE07	EUR	€905,068.00	$724,054.40	CA$633,547.60	$543,040.80
STORE08	EUR	€870,414.00	$696,331.20	CA$609,289.80	$522,248.40
STORE09	EUR	€895,304.00	$716,243.20	CA$626,712.80	$537,182.40
STORE1	EUR	€959,588.00	$767,670.40	CA$671,711.60	$575,752.80

Currency results:
1. The local currency property value for Store02 is EUR.
2. The local currency is EUR. This value is stored in the database.
3. Group currency is USD. This value is stored in the database.
4. Alternative currency is calculated but not stored. This is a model 'currency conversion measure'.
5. Simulated currency is a story calculated measure.

Figure 53: Currency translation results in a story

Note:
Displaying the local currency (property) in the columns is optional.

In the figure entitled *Currency Translation Results in a Story*, the new model is demonstrated. You can access the example above in the U00S_Currency story in the SACP20 → Solutions folder.



In order to create a converted measure in the story:
1. Add a Measure for Currency Conversion.
2. Name the new column.
3. Select the target currency: either fixed or based on the entity currency.
4. Select a category either dynamically, or fixed to actual or plan.
5. Choose the source measure.
6. Select the date to determine the exchange rate.

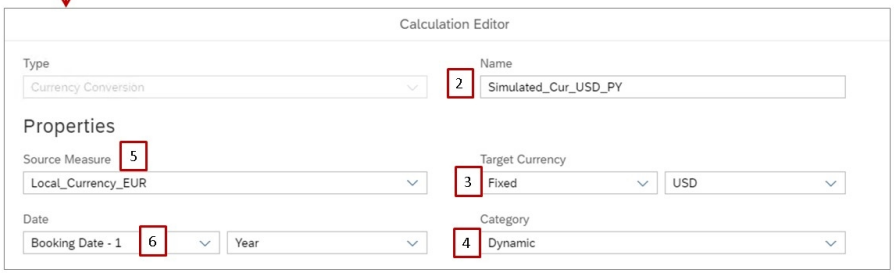


Figure 54: Using story conversion measures

Note:
The *Booking Date*, for example, is determined through the date dimension (specified in the currency setting) in the model.

If a currency conversion is added and there are no exchange rates, the system will direct you to the currency table (to add the missing rates).

Currency translation table

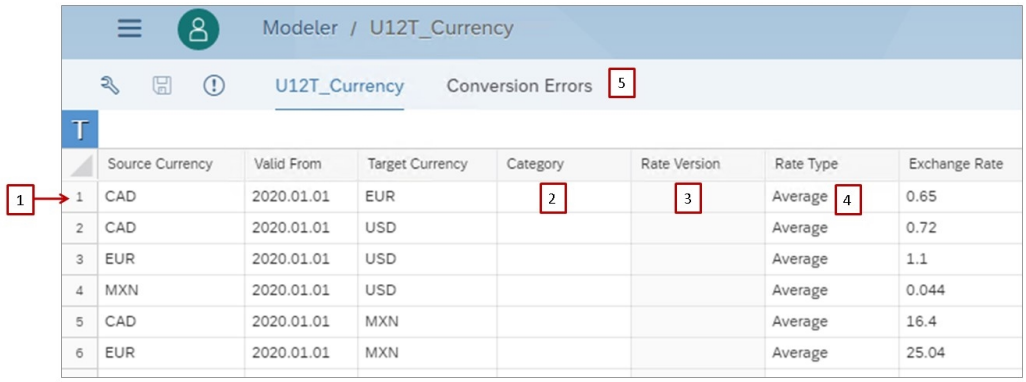


Figure 55: Translation table

Currency conversion tables are defined independently of models. You can apply a selected table to any model you create. Currency tables are listed on a separate *Currency Conversion* page, where you can add, copy, and delete currency tables as required.

Currency Conversion Tables https://help.sap.com/viewer/00f68c2e08b941f081002fd3691d86a7/release/en-US/9348d55977e94994ab27bf031389d8f2.html

Model and dimension currency settings

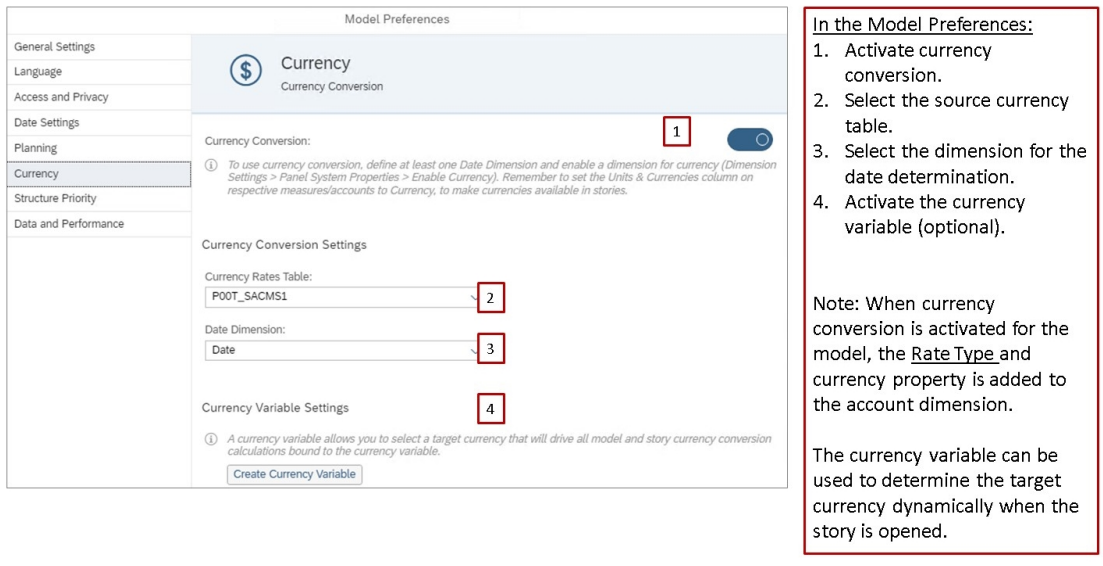


Figure 56: Model preferences - currency related

If the currency variable is turned on, the variable can be used to determine the target currency for a translated measure (but not in a measure).

- One of the main use cases involves SAP Analytics Cloud models designed for SAP ERP financial data. In this case the following measures would be used:
  - Local currency amount
  - Group currency amount
  - Others if needed
  - Quantity
- Measures can also be added to the model that calculate currency values on-the-fly.

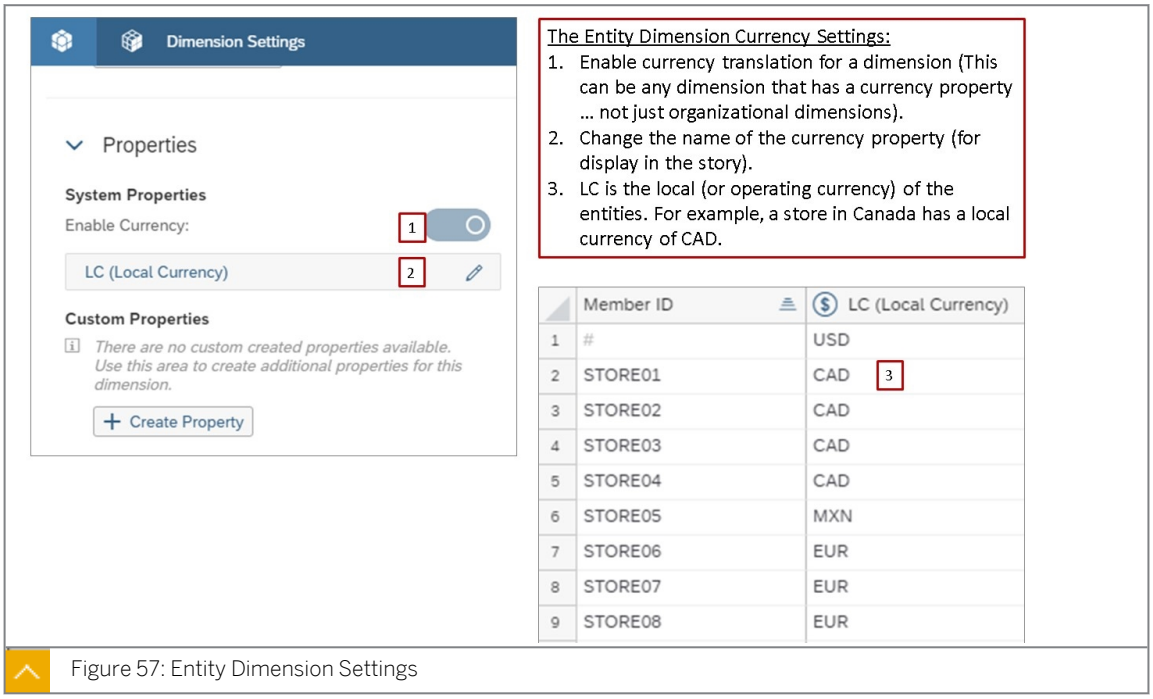


Figure 57: Entity Dimension Settings

Switch on *Enable Currency* to add a *Currency* column to the dimension. Using three-character currency codes, this column identifies the source currency for data that belongs to each leaf member. The *Currency* column is enabled by default for the *Organization* dimension, and can also be added to generic dimensions.

For each converted measure in the model, a dimension (any dimension with currency enabled) is set as the currency dimension.

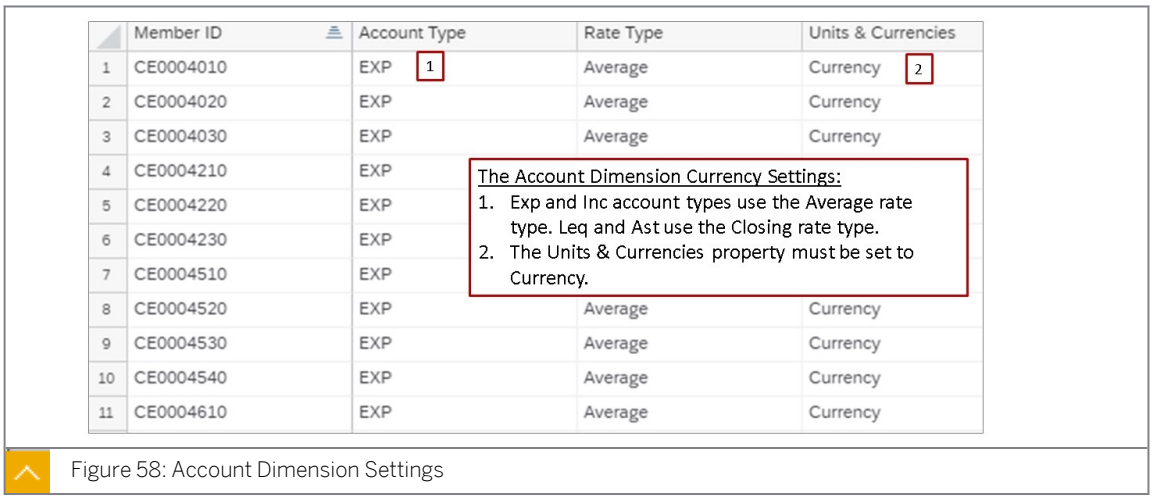


Figure 58: Account Dimension Settings

In the account dimension, the *Rate Type* attribute column appears when you switch on *Currency Conversion* in the model preferences. With currency conversion switched on, set the *Rate Type* to *Average* for INC and EXP accounts, and to *Closing* for AST and LEQ accounts. This setting corresponds to the rate type for exchange rates in your currency table. It lets you distinguish between the average exchange rate over a period and the closing rate at the end of the period.

Only the average and closing rate types are possible. Custom rate types are not possible.

Currency conversion examples

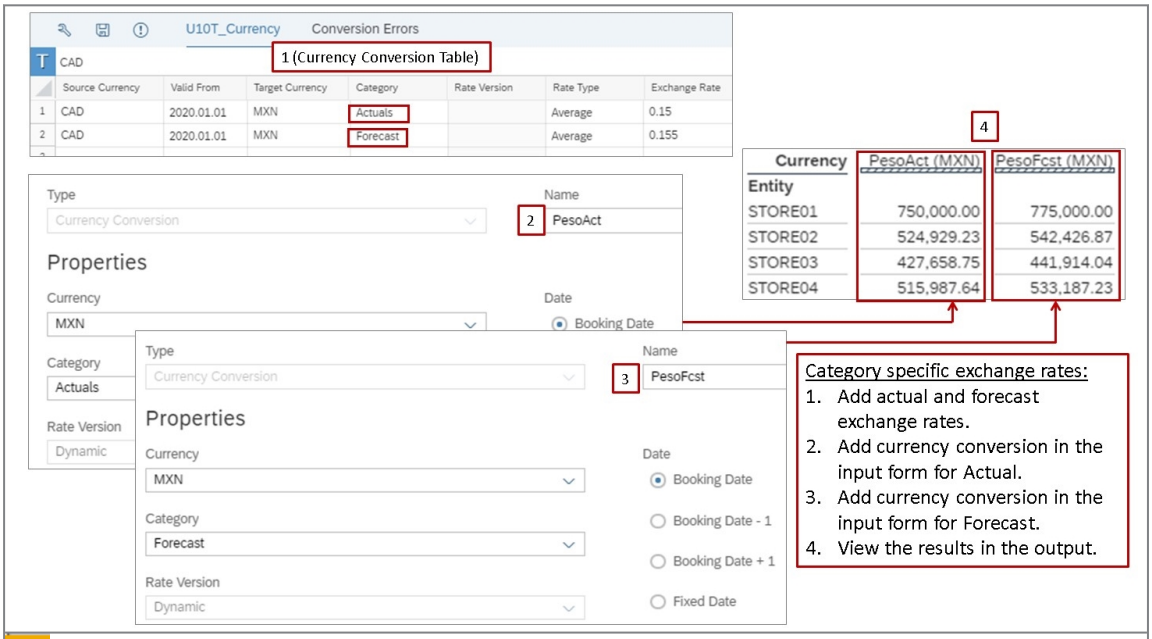


Figure 59: Forecast vs actual exchange rates

Note:
The CAD value for Store 1 in the preceding image, is 5,000,000:

- 5,000,000 CAD x .15 = 750,000 MXN
- 5,000,000 CAD x .154 = 750,000 MXN

If needed, you can compare values that use exchange rates for different *categories*. This is useful for what-if scenarios.

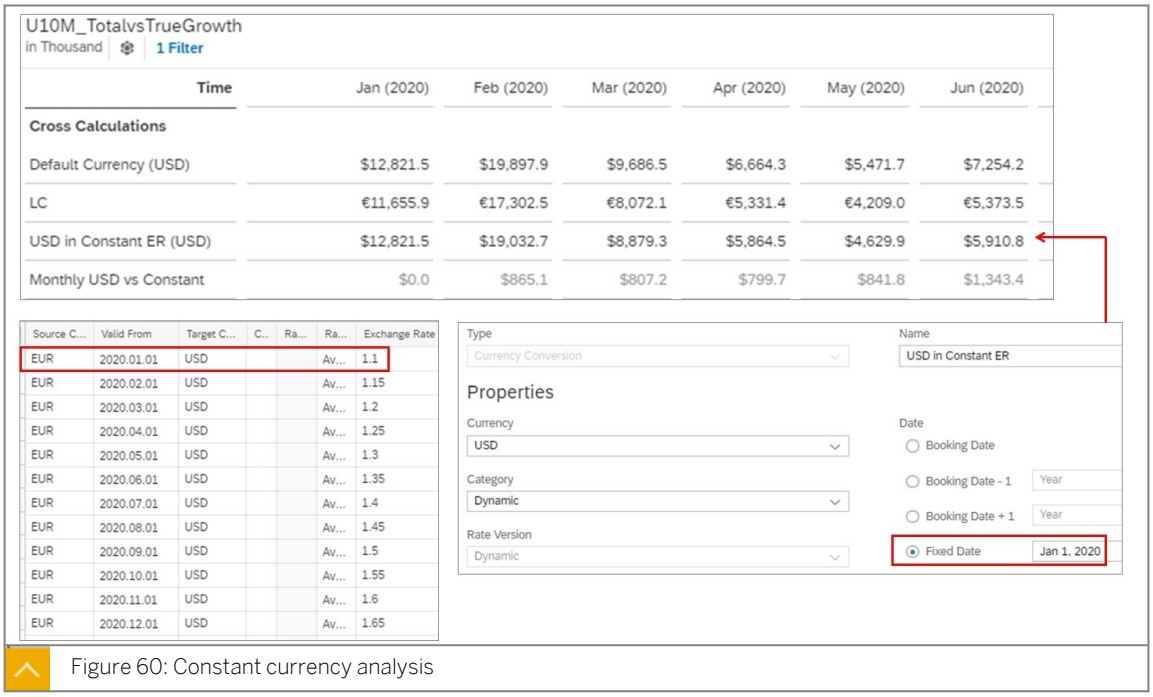


Figure 60: Constant currency analysis

Constant currency scenario: I want to see companies annual growth in a model currency (such as USD) but I want a flat conversion rate applied so I filter out the currency conversion impact on my figures.

In the preceding figure:

In the example below:

- The *Default Currency (USD)* values are translated with the monthly average exchange rates for 2020. For example:
  - Jan: 11,655.9 x 1.1 = 12,821.5
  - Feb: 17,302.5 x 1.15 = 19,897.9
- The *USD in Constant ER (USD)* values are translated with the Jan average exchange rate for 2020. For example:
  - Jan: 11,655.9 x 1.1 = 12,821.5
  - Feb: 17302.5 x 1.1 = 19,032.7

LESSON SUMMARY

You should now be able to:

- Configure and translate currencies

Unit 3
Lesson 6

Designing and creating time hierarchies

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Explain the prerequisites required for creating hierarchies

Build a custom time hierarchy

The hierarchies you build depend on the properties available in the date dimension. For user-managed date dimensions, you must have at least one hierarchy.

Once the properties have a semantic type assigned, you can add them as levels within the hierarchy. Make sure to follow these guidelines:

- The hierarchy has the mandatory Year semantic type as the highest level.
- The semantic types within the hierarchy are ordered from the biggest to the smallest time unit. For example, Month must be below Quarter, and above Week.
- Each semantic type within the hierarchy is only used once, except for the Other semantic type.
- If the hierarchy uses the semantic type Month, do not include the semantic type Period, and vice versa.
- The lowest level of the hierarchy is the ID property if there is one. Otherwise, it must be a unique property.
- Description properties cannot be used in the hierarchy.

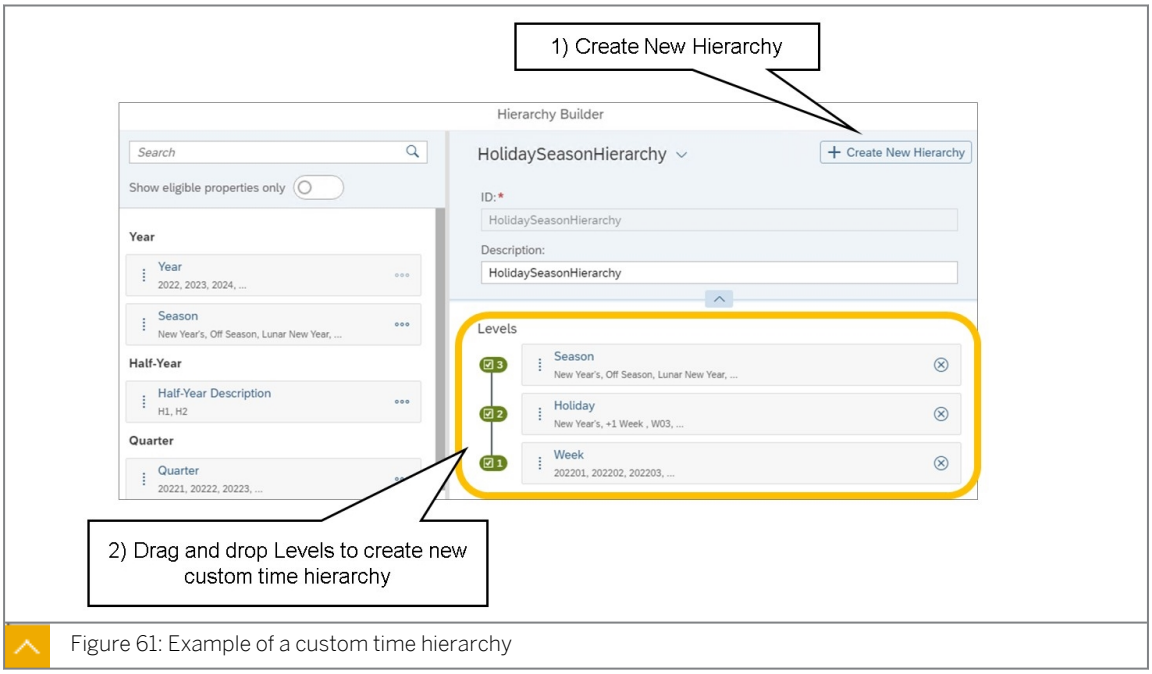


Figure 61: Example of a custom time hierarchy

In the preceding *Building a Custom Time Hierarchy* figure, the hierarchy has 3 levels:

- Level 3: Season
- Level 2: Holiday
- Level 3: Week

> Note:
> The seasonal hierarchy that we are illustrating is just one example of how custom time hierarchies can be used.

Figure 62: Using a custom time hierarchy in a story

The date hierarchies provide many options when it comes to analyzing sales over time. For example, date can be displayed in the rows of a table using a custom seasonal time hierarchy. In addition, the season and holiday custom properties can also be displayed in the columns if needed.

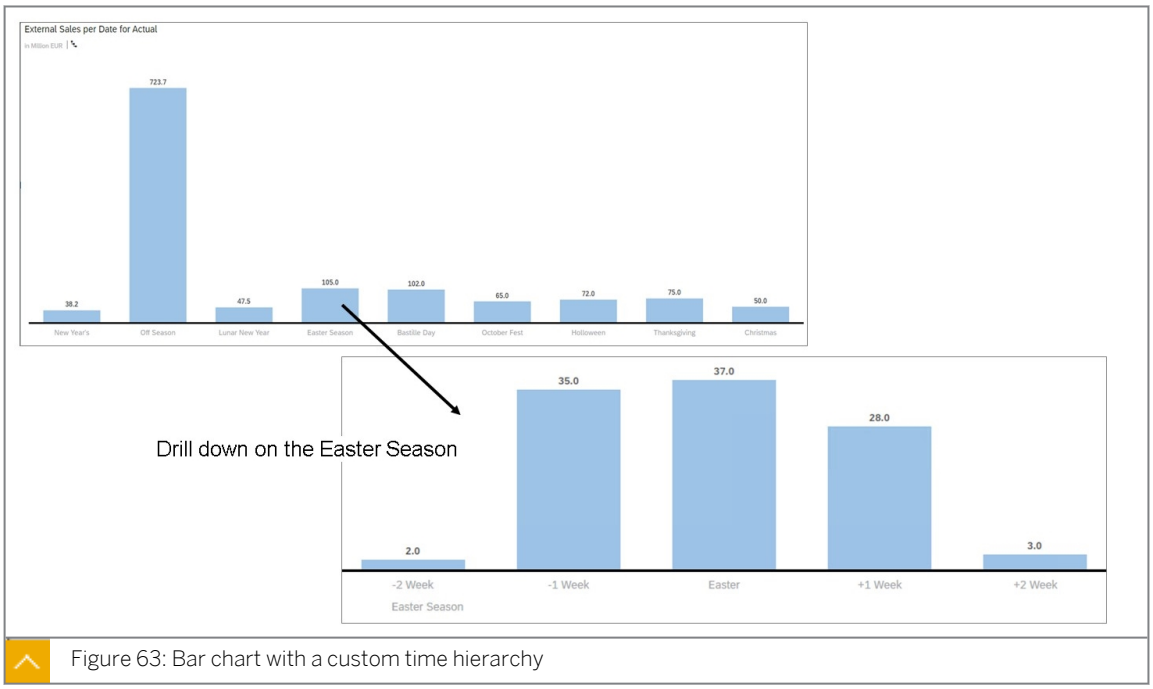


Figure 63: Bar chart with a custom time hierarchy

The *Bar Chart with Custom Time Hierarchy* figure is one of the end results. It displays sales values for each season. This meets a business requirement for many consumer products companies.

LESSON SUMMARY

You should now be able to:

- Explain the prerequisites required for creating hierarchies

Learning Assessment

1. Models always contain data.

   Determine whether this statement is true or false.

   - [ ] True
   - [ ] False

2. What are the types of models used in SAP Analytics Cloud? Note: there are two correct answers for this question.

   Choose the correct answers.

   - [ ] **A** Planning
   - [ ] **B** Story specific
   - [ ] **C** Analytic
   - [ ] **D** Application specific

3. What dimension types can you add when you create a model in SAP Analytics Cloud? Note: there are three correct answers for this question.

   Choose the correct answers.

   - [ ] **A** Account
   - [ ] **B** Version
   - [ ] **C** Generic
   - [ ] **D** Organization
   - [ ] **E** Date

4. The New Model requires an Account dimension.

   Determine whether this statement is true or false.

   - [ ] True
   - [ ] False

5. Data wrangling involves manipulating the data to make it accurate and to match your business needs.

   Determine whether this statement is true or false.

   - [ ] True
   - [ ] False

6. Flat file imports can be scheduled.

   Determine whether this statement is true or false.

   - [ ] True
   - [ ] False

7. You can change field selections when importing from an SAP Universe.

   Determine whether this statement is true or false.

   - [ ] True
   - [ ] False

8. You can select the data range scope when using the clean and replace import method.

   Determine whether this statement is true or false.

   - [ ] True
   - [ ] False

9. If just one step in a scheduled import fails, all steps fail.

   Determine whether this statement is true or false.

   - [ ] True
   - [ ] False

10. Dimension member formulas are only available for the account type dimension.

    Determine whether this statement is true or false.

    - [ ] True
    - [ ] False

11. Model-specific calculated measures have an account type property that is used to swap signs.

    Determine whether this statement is true or false.

    ☐ True

    ☐ False

12. Where does SAP Analytics Cloud store exchange rates?

    Choose the correct answer.

    ☐ **A** In a dataset

    ☐ **B** In a currency table

    ☐ **C** In a model

    ☐ **D** In a dimension

13. What time pattern is supported for the time dimension?

    Choose the correct answer.

    ☐ **A** 4-4-4

    ☐ **B** 4-5-5

    ☐ **C** 13x4

    ☐ **D** 12x4

Learning Assessment - Answers

1. Models always contain data.

   Determine whether this statement is true or false.

   - [ ] True
   - [X] False

   Correct. Live models do not contain data.

2. What are the types of models used in SAP Analytics Cloud? Note: there are two correct answers for this question.

   Choose the correct answers.

   - [X] A Planning
   - [ ] B Story specific
   - [X] C Analytic
   - [ ] D Application specific

   Correct. Planning and Analytic models are the model types used in SAP Analytics Cloud. Models are always public, they are not story or application specific.

3. What dimension types can you add when you create a model in SAP Analytics Cloud? Note: there are three correct answers for this question.

   Choose the correct answers.

   - [X] A Account
   - [ ] B Version
   - [X] C Generic
   - [X] D Organization
   - [ ] E Date

   Correct. When creating a model, you can add account, generic, and organization dimension types. Version and date dimensions are added by the system.

4. The New Model requires an Account dimension.

   Determine whether this statement is true or false.

   ☐ True

   ☒ False

   Correct. The Account dimension is optional in the New Model.

5. Data wrangling involves manipulating the data to make it accurate and to match your business needs.

   Determine whether this statement is true or false.

   ☒ True

   ☐ False

   Correct. Data wrangling involves manipulating the data to make it accurate and to match your business needs.

6. Flat file imports can be scheduled.

   Determine whether this statement is true or false.

   ☐ True

   ☒ False

   Correct. Flat file imports cannot be scheduled.

7. You can change field selections when importing from an SAP Universe.

   Determine whether this statement is true or false.

   ☐ True

   ☒ False

   Correct. You cannot change field selections when importing from an SAP Universe.

8. You can select the data range scope when using the clean and replace import method.

   Determine whether this statement is true or false.

   ☒ True

   ☐ False

   Correct. You can select the data range scope when using the clean and replace import method.

9. If just one step in a scheduled import fails, all steps fail.

   Determine whether this statement is true or false.

   ☐ True

   ☒ False

   Correct. if just one step in a scheduled import fails, the subsequent steps can run if the import allows it.

10. Dimension member formulas are only available for the account type dimension.

    Determine whether this statement is true or false.

    ☒ True

    ☐ False

    Correct. Dimension member formulas are only available for the account type dimension.

11. Model-specific calculated measures have an account type property that is used to swap signs.

    Determine whether this statement is true or false.

    ☐ True

    ☒ False

    Correct. Only account dimensions have an account type property that is used to swap signs.

12. Where does SAP Analytics Cloud store exchange rates?

    Choose the correct answer.

    ☐ A In a dataset

    ☒ B In a currency table

    ☐ C In a model

    ☐ D In a dimension

    Correct. Exchange rates are stored in a separate currency table.

13. What time pattern is supported for the time dimension?

Choose the correct answer.

- [ ] A 4-4-4
- [ ] B 4-5-5
- [x] C 13x4
- [ ] D 12x4

Correct. The supported weekly time patterns are: 4-4-5, 4-5-4, 5-4-4, 13x4.

UNIT 4 Creating live models

UNIT OBJECTIVES

- Access data for your model with the data analyzer
- Create a live model with S/4HANA data
- Create a live model with with HANA data
- Create a live model with SAP Business Warehouse data
- Create a live model using SAP BusinessObjects data

Unit 4
Lesson 1

Using Data Analyzer to preview your data structure

LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Access data for your model with the data analyzer

Data structures for live models

You can access the data from a live connection without creating a model by using the data analyzer.

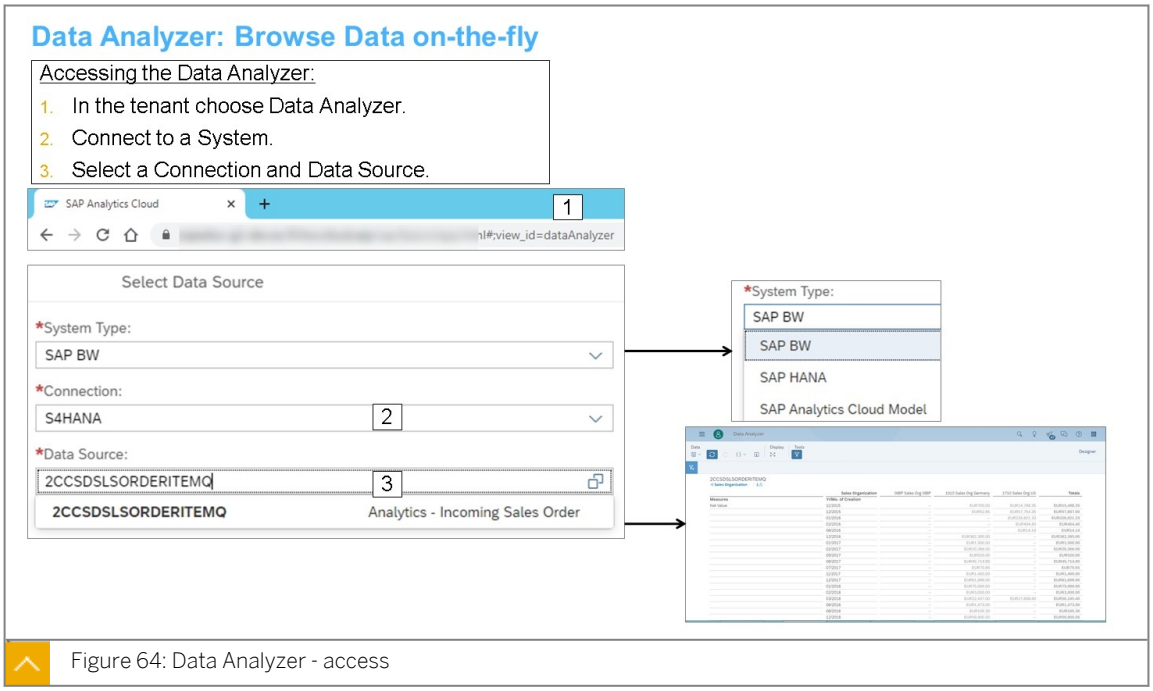


Figure 64: Data Analyzer - access

Data analyzer is a predefined ready-to-run service for SAP Business Warehouse (SAP BW) queries, SAP HANA Live views, and SAP Analytics Cloud models for ad-hoc analysis. It uses SAP BW live connections, SAP HANA views and models created in SAP Analytics Cloud.

All SAP BW queries, SAP HANA Live views and models can be accessed directly in the Select Data Source dialog and no additional model is created.

Data analyzer user interface

1. Bullet-proof user interface.
2. Good for quick, Ad-Hoc inquiries.
3. Drag & drop | Pause refresh | Swap axis Filtering.
4. Save to your results as an “Insight” file.

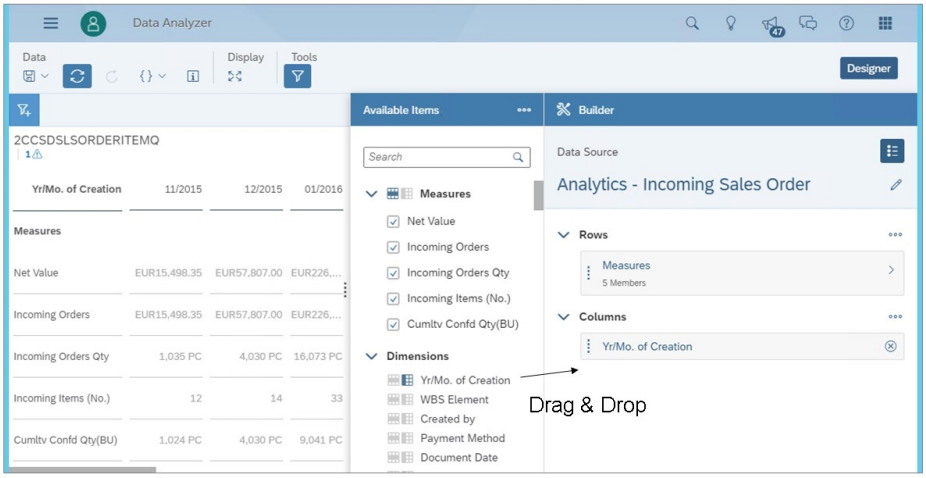


Figure 65: Data Analyzer - User Interface

Data analyzer contains a table, a filter area, and a builder panel with navigation capabilities to add and remove dimensions and measures from the table. In addition, you find a menu bar with a Refresh option and an Edit Prompts dialog (in case your data source is designed for setting prompts).

Table formulas are available but calculated measures are not.

After you have drilled down to the data details according to your needs and analyzed your data, you might want to save this insight. For this, choose Save in the upper left corner of data analyzer. In the Save Insight dialog, select the file location (public or private) for your insight and enter a name and description for it.

Note:
You can also test a CDS view from S/4HANA. In the data analyzer, connect to BW → S/4 → Log in with wss4 / Welcome1 → search for journal → If prompted, use company code 1010. In Fiori, this is the Journal Entry Analyzer. Launch Fiori from the SACP30 folder in the N drive with user s4h00-00 / Welcome1. It can be accessed with the Query Browser. The technical name of the view is C_GLLineItemsQ0001.

LESSON SUMMARY

You should now be able to:

- Access data for your model with the data analyzer

Unit 4
Lesson 2

Creating live models with S/4HANA data

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Create a live model with S/4HANA data

SAP S/4HANA integration

SAP S/4HANA is a complete enterprise resource planning (ERP) system with built-in intelligent technologies, including AI, machine learning, and advanced analytics. It helps companies adopt new business models, manage business change at speed, orchestrate internal and external resources, and use the predictive power of AI.

With SAP Analytics Cloud's ability to connect to SAP S/4HANA, S/4HANA customers can leverage their on-premise system with SAP Analytics Cloud's cloud-based modeling and analytics capabilities.

The Integration with an S/4HANA system can be done in two ways:

- **As a live data connection:** no data will be stored on the cloud.
- **As an import data connection:** data will be stored in the cloud and updated periodically.

Note:
When using SAP Analytics Cloud as the interface for planning activities, data must be imported from S/4HANA.

Identify data for S/4HANA models

SAP S/4HANA live data connection

You can create live data connections to SAP S/4HANA systems using the direct, tunnel, and cloud connection types. You can build stories based on released CDS views or leverage queries created by you.

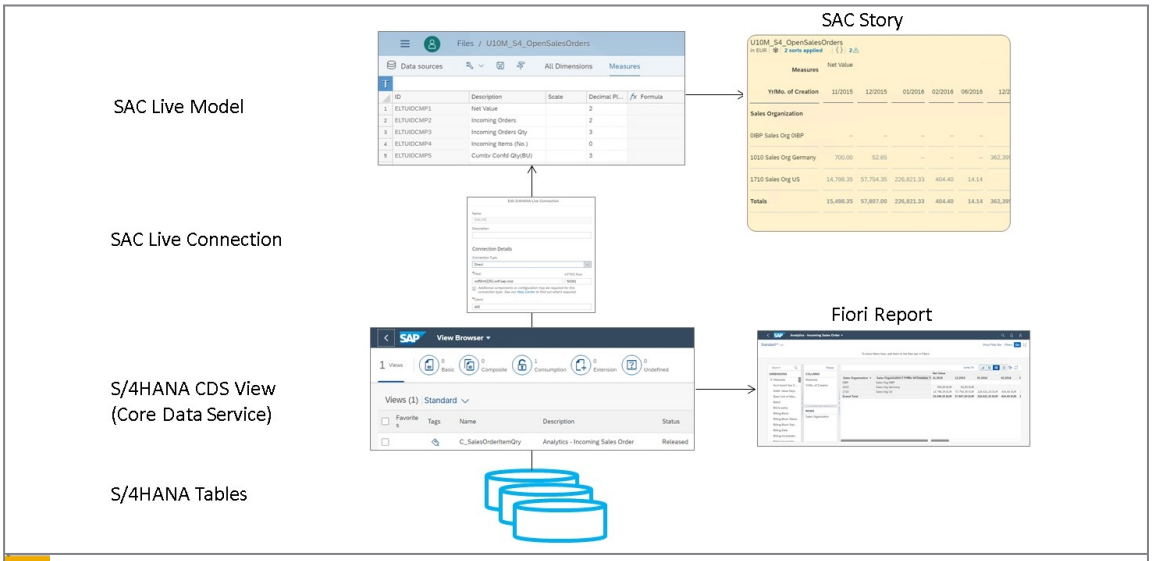


Figure 66: SAP S/4HANA live data flow

In the figure *SAP S/4HANA Live Data Flow* we illustrate the data flow from S/4HANA to SAP Analytics Cloud. The *View Browser* Fiori app is used to access CDS views of all kinds.

Note:
Use the *Access Live SAP S/4HANA Data with a Model* procedure to access the View Browser app.

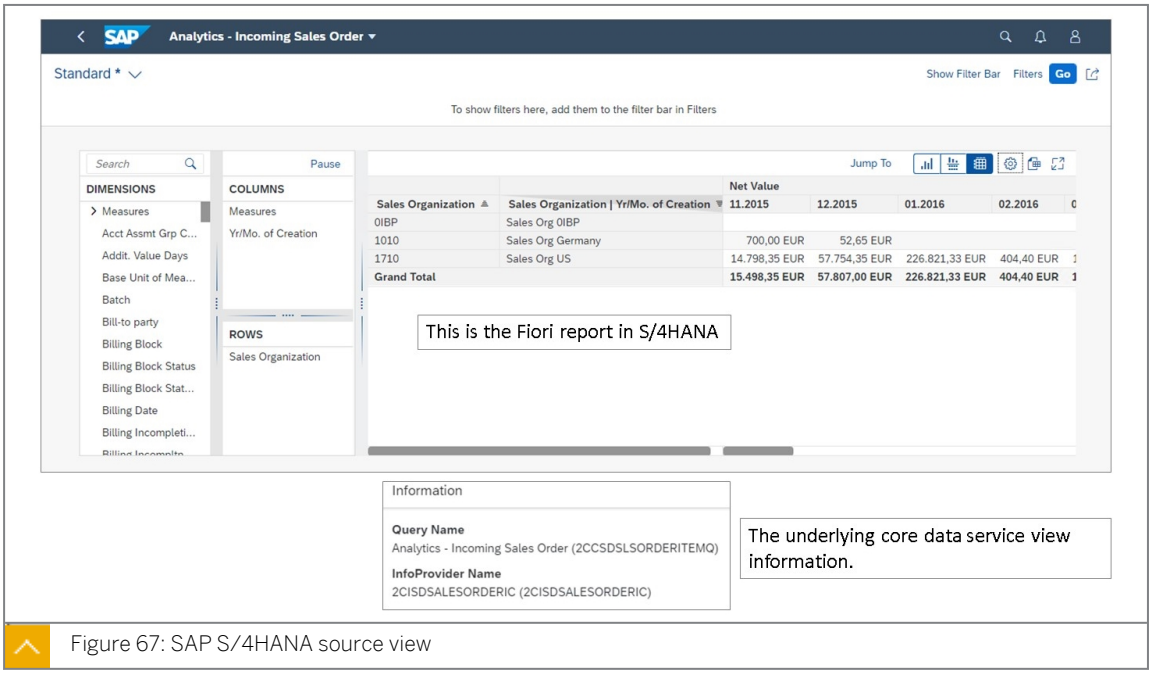


Figure 67: SAP S/4HANA source view

LESSON SUMMARY

You should now be able to:

- Create a live model with S/4HANA data

Unit 4
Lesson 3

Creating live models with HANA data

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Create a live model with with HANA data

SAP HANA integration

SAP HANA and SAP HANA Cloud

SAP Analytics Cloud is integrated with SAP HANA (on premise) and SAP HANA Cloud as follows:

- Live connections:
  - Available for SAP HANA
  - Available for SAP HANA Cloud
- Import connections: Available for SAP HANA

In SAP Analytics Cloud, you can create live data connections to SAP HANA Cloud using Live Connection with Connection Type “SAP HANA Cloud”. SAP Analytics Cloud can delegate model management to SAP HANA Cloud by building SAC models using HANA Calculation Views as a data source. Calculation Views allows users to define more advanced slices on the data available in the SAP HANA database. As a live data connection, no data will be stored on the cloud.

SAP HANA Multi-Dimensional Services (MDS) is used to process multidimensional queries including aggregation, transformation and calculations in SAP Analytics Cloud, SAP HANA Cloud, and SAP HANA on-premise.

SAP Analytics Cloud models based on HANA Calculation View can improve stories performance by delegating and moving data intensive calculations to SAP HANA Cloud. In this case, SAP Analytics Cloud is the application layer in the Future approach scenario as specified in diagram. By pushing down data intensive calculations to the SAP HANA Cloud and only returning the result, this would optimize the performance of the SAP Analytics Cloud tenant.

Identify data for SAP HANA models

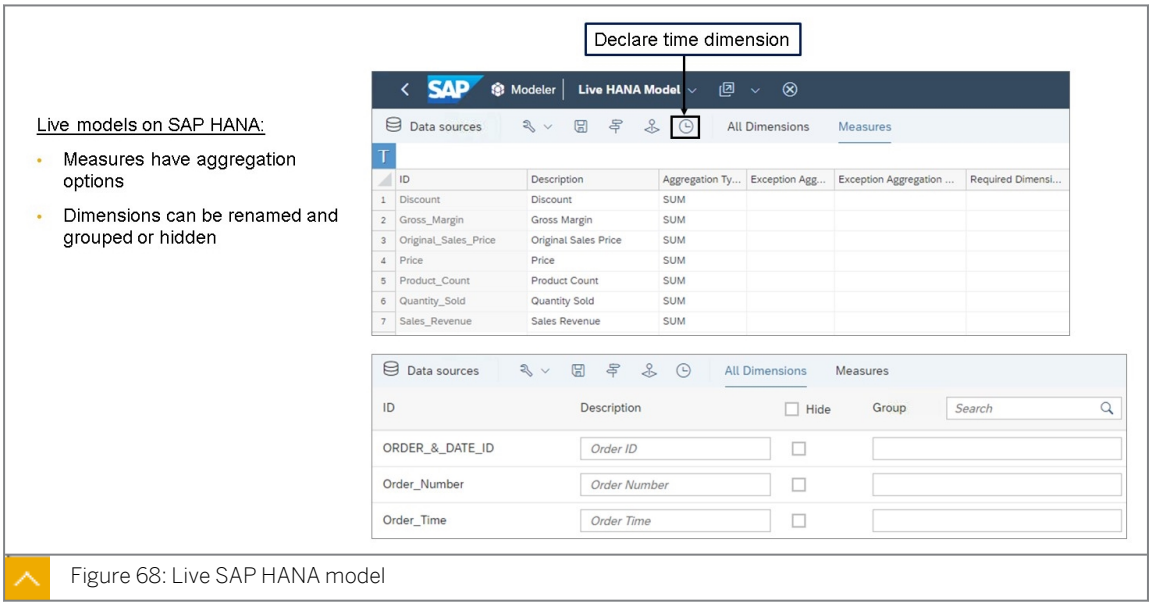


Figure 68: Live SAP HANA model

When working with live models for SAP HANA, there are two unique features:

1. The measures' aggregation settings can be adjusted in the live model.
2. A time dimension can be declared, allowing for the use of the automatic time-determination features in stories.

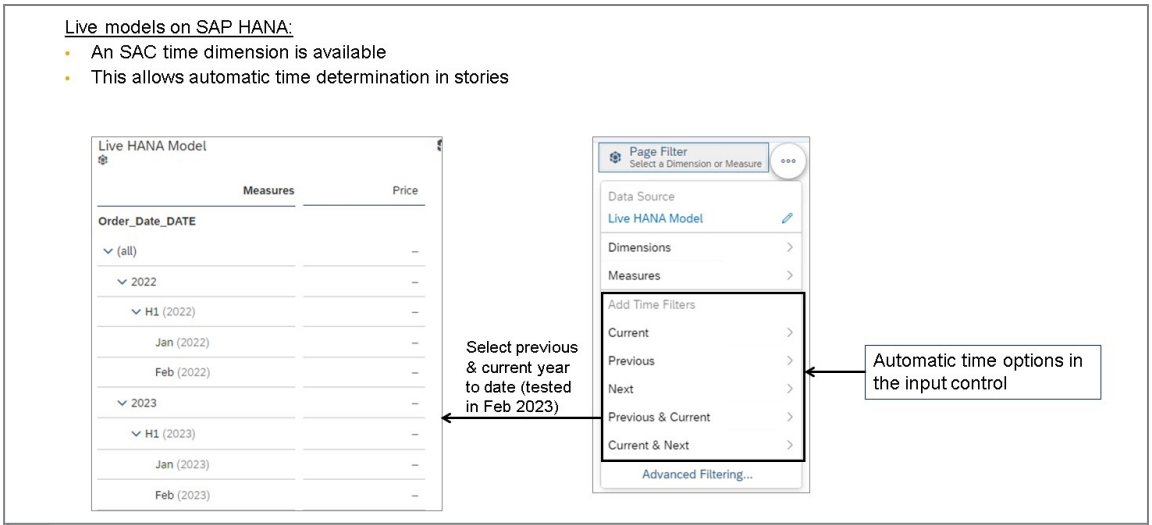


Figure 69: Live SAP HANA model - use automatic time-determination in stories

To access live SAP HANA Cloud data

Note:
This procedure is only for the instructor (SSG setup instructions required prior to demonstration).

You need to access live data from SAP HANA Cloud using a Calculation View.

1. Start the *SACDEMO* instance in *HANA Cloud Central*.
   a) Open Google Chrome and enter the *HANA Cloud Central* URL: https://hana-cockpit.cfapps.eu11.hana.ondemand.com/hcs/sap/hana/cloud/index.html#/org/3e841fc3-64b2-4651-b26f-d505a5ab9d19/space/1b1f3f94-1547-4be4-be3c-6c86d0a426e3/databases?databaseguid=32c217f9-64c8-4417-815d-527939a69c61

   b) Sign in to the default identity provider with the User and Password as specified in the SSG.

   c) The SACDEMO should be running. If not, choose ...Actions → Choose Start.

2. Access the Business Application Studio.
   a) Open a new Google Chrome window, enter URL: https://hana4sac.eu11cf.applicationstudio.cloud.sap/index.html

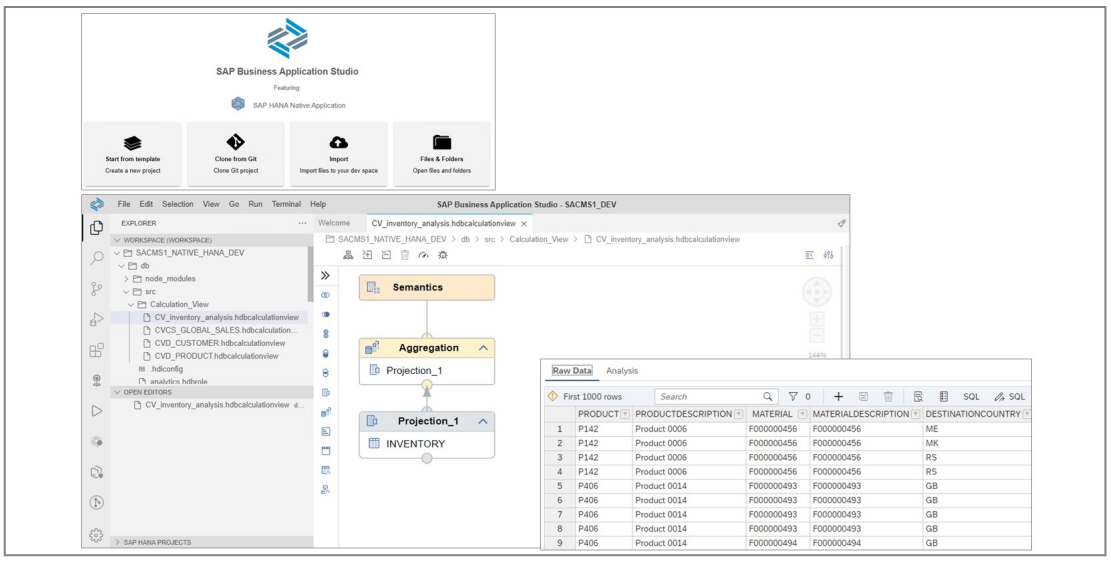
   b) Next to SACMS1_DEV, choose  *Start* → Wait a few seconds until it is *RUNNING*.

3. Click on the *SACMS1_DEV* space.

4. From the *WORKSPACE* on the left, navigate: db → scr → Calculation View.

5. Double-click *CV_inventory_analysis.hdbcalculationview*.

6. Right-click the Aggregation node → select *Data Preview*.



Result

Note:
If you see an error in the Business Application Studio: *Failed to get Cloud Foundry token. Please log back in to re-authenticate and reopen the editor.* On the lower right, choose the hyperlink: *Log in to Cloud Foundry.* The link https://api.cr.eu11.hana.ondemand.com appears. Press Enter. Enter user and password provided in the SSG. Select the default organization *KTE_HANA_hana4sac* and space *SACDC*. Re-open the *Business Application Studio* editor. Re-open the *Business Application Studio* editor.

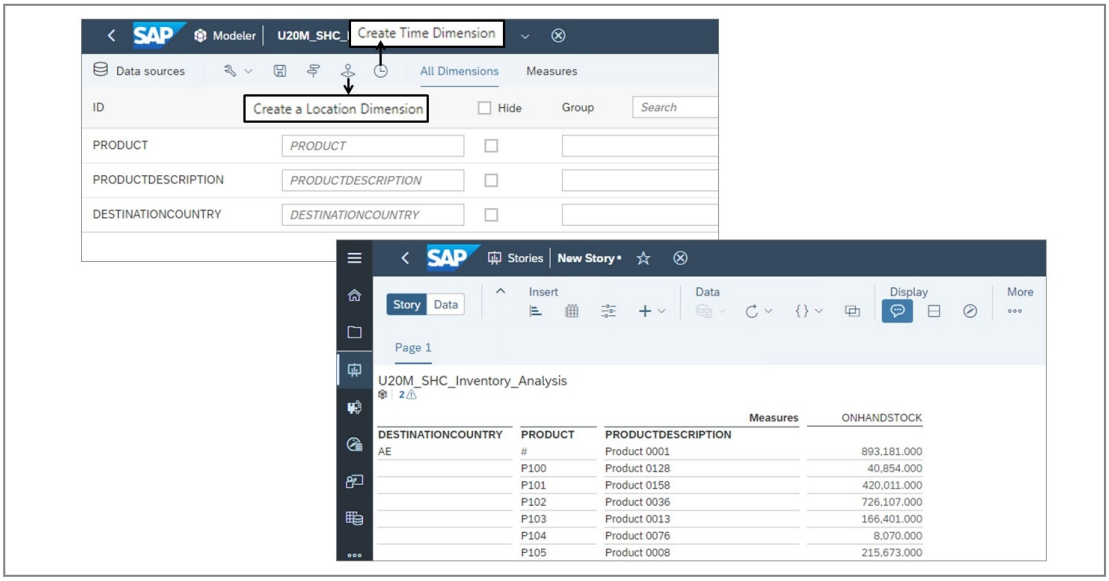
7. Now it is time to create an SAC Model to access this data. Go to the SAC browser with the A00 user and password provided in the SSG.
8. On the upper left, choose the *Expand Navigation Bar* icon.
9. Create a model from a live data source.
   a) From the Navigation Bar choose the *Modeler* icon.
   b) Choose *Live Data Model*.
10. Select the live data connection and inventory calculation view.
   a) Select System Type *SAP HANA* → Select Connection *HANACLOUD*.
   b) Select Data Source: *CV_inventory_analysis*.

Result

> Note:
> If the underlying calculation view contains a time field, the time dimension in SAC can be configured and thereby allow automatic time determination in stories. If the underlying calculation view contains a latitude and longitude field, the location dimension in SAC can be configured and thereby allow geo mapping.

11. Save the model as **`U##M_SHC_Inventory_Analysis`**.
12. Create a story with a table to view the data.
   a) Go to Actions → Story → Canvas.
   b) Choose Table.
   c) Add Product to the rows.

Result



13. Close the extra tabs for SAP HANA Cloud.

LESSON SUMMARY

You should now be able to:

- Create a live model with with HANA data

Unit 4
Lesson 4

Creating live models with SAP Business Warehouse data

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Create a live model with SAP Business Warehouse data

SAP Business Warehouse Integration

SAP Analytics Cloud and SAP BW, SAP BW/4HANA

SAP Analytics Cloud integrates seamlessly with SAP Business Warehouse (SAP BW) to make the most of your existing investments. And the Integration of an SAP Business Warehouse can be done in two ways:

- As a **live data connection:** no data will be stored in SAP Analytics Cloud.
- As an **import data connection:** data will be stored in SAP Analytics Cloud and updated periodically.

Supported SAP BW elements in live models

The following SAP BW elements are supported in SAP Analytics Cloud stories:

- Time-dependent hierarchies
- BW Variables
- BW Variants and Personalization
- BW Structures
- Parallel execution of BW queries

BW hierarchy support

Figure 70: Time dependent hierarchy support

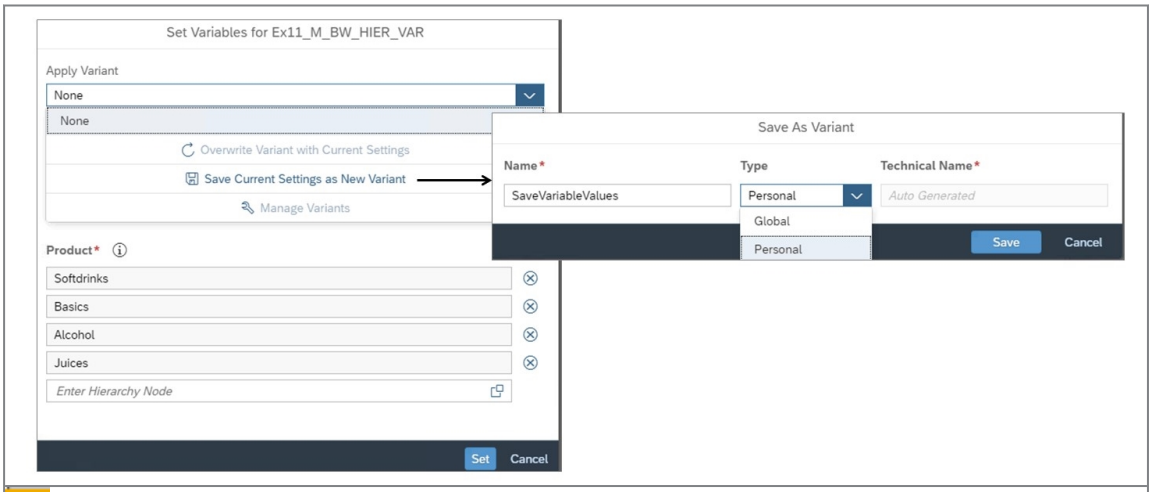


Figure 71: Creating variants

Variants are available for any live model that has prompts.

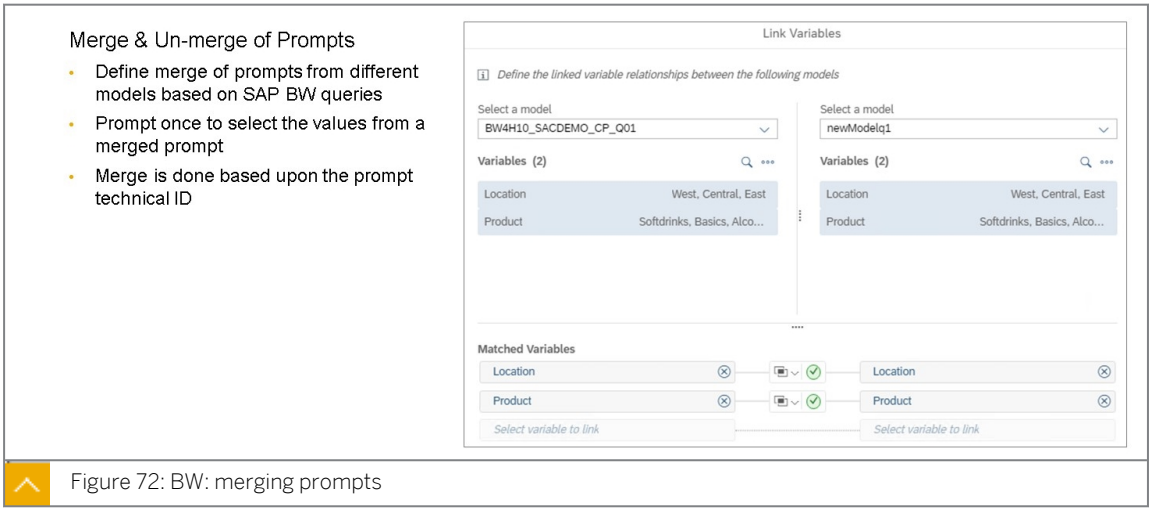


Figure 72: BW: merging prompts

BW: variable support

- Exit and authorization variables are supported
- Operators: you can define range exclusions on dimension members
- Show booked values in the search help based on the setting in the BW query

BW Structures: Easily visualize line of business data from BW structure concepts

- Leverage BW queries with two structures
- Calculated value of cell definitions created with the cell editor in BW query is supported

	Measures	Sales_Value	Quantity
Location	Struct.		
	Q2 2022	$2,073,447.20	1,446,748.00
	Q1 2023	$1,994,433.00	1,174,110.00
	Q2 2023	$2,137,373.20	1,484,408.00
	Change Q2 vs. Q1 (%)	7.17 %	26.43 %
> California	Change Q2 2023 vs. Q2 2022 (%)	3.08 %	2.60 %
	Q2 2022	$2,073,447.20	1,446,748.00
	Q1 2023	$1,994,433.00	1,174,110.00
	Q2 2023	$2,137,373.20	1,484,408.00

Figure 73: BW: structures support

In addition, live connectivity makes it easy to visualize line-of-business data from SAP BW structure concepts, as shown in the figure, Live Data Connectivity to SAP BW: BW Structures Support.

Parallel processing of BW queries

BI administrators can configure the number of parallel sessions allowed on your SAP BW system.

Stories containing multiple BW queries will be executed in parallel.

Recommended guidelines

Parallel sessions will introduce additional overhead on SAP BW server resources. Ensure BW server is properly sized to handle increase in session processing

Only use this option where performance improvement is absolutely required. Manage your server resources judiciously.

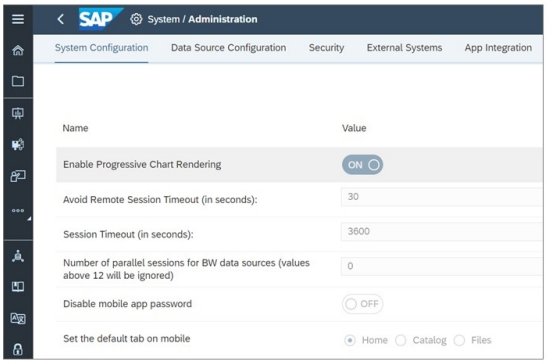


Figure 74: Parallel processing to enhance story performance

Note:
Display and navigation attributes are not fully supported. The end user must manually change the attribute column in SAC as a property classification.

For more information on SAP BW connections, see: https://help.sap.com/viewer/00f68c2e08b941f081002fd3691d86a7/release/en-US/5dd668839c9f44599d61e6f0016dd553.html

Identify data for Business Warehouse models

Import data connection

When using an imported data connection to SAP BW, you have the following advantages, as it:

- Uses existing SAP BW queries to securely import your data into the cloud.
- Applies SAP Analytics Cloud native data security concepts.
- Leverages the calculation engine of SAP HANA to allow for additional analytic use cases.

Live data connection

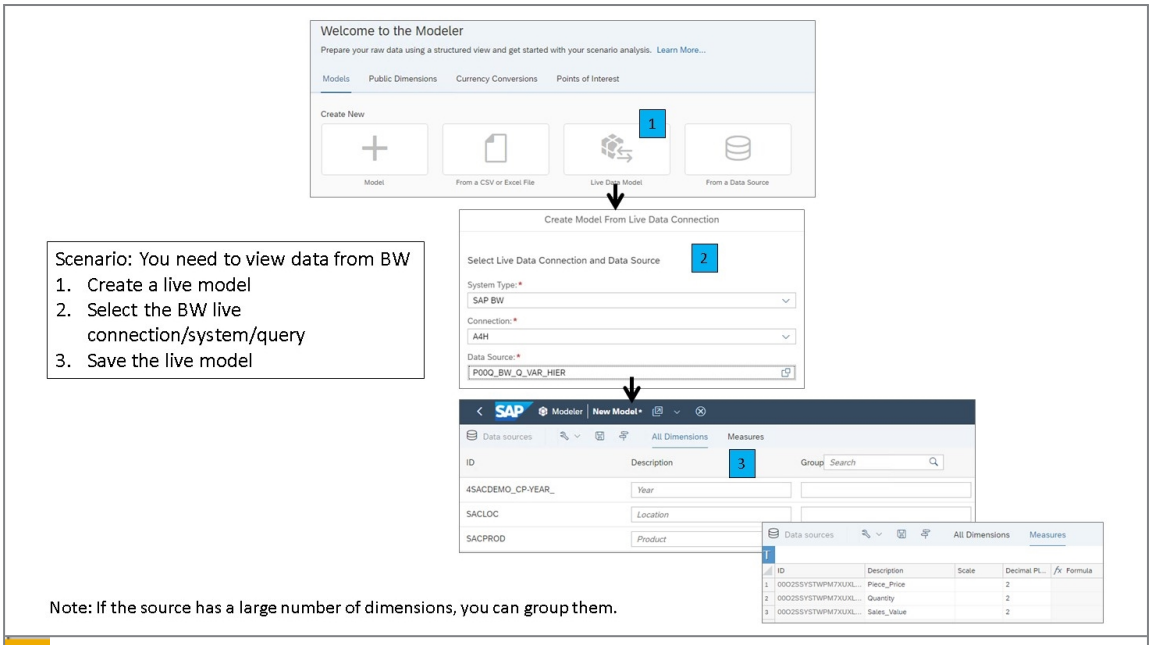


Figure 75: BW live model workflow

When using a live data connection to SAP BW, you have the following advantages:

- Accesses SAP BW features that cannot be accessed through standard SQL or MDX query interfaces.
- Leverages the SAP BW metadata without additional modeling.
- Re-uses existing authorization concepts.
- Connects to SAP BW, as well as generated SAP HANA views.

> Note:
> For more information on supported features and required updates for the integration of SAP BW with SAP Analytics Cloud, see: SAP Note - 2541557

> Note:
> When creating linked dimensions for SAP BW data sources, you can choose to link on matching hierarchies.

LESSON SUMMARY

You should now be able to:

- Create a live model with SAP Business Warehouse data

Unit 4
Lesson 5

Creating live models with BusinessObjects data

LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Create a live model using SAP BusinessObjects data

SAP BusinessObjects integration

The SAP BusinessObjects Business Intelligence platform is a flexible and scalable solution for delivering information to end users, in multiple forms including dashboards and interactive reports, via any web application or corporate portal. The platform delivers tangible benefits extending across and beyond the organization, as an integrated suite for reporting, analysis, and information delivery. It also provides a solution for increasing end-user productivity and reducing administrative efforts. For example, it is used to distribute weekly sales reports, to provide customers with personalized service offerings, or to integrate critical information in corporate portals.

SAP BusinessObjects Business Intelligence on-premise solutions work with SAP Analytics Cloud in a hybrid analytics scenario. In the long term, SAP has strategically positioned SAP Analytics Cloud as the augmented analytics tool. Meanwhile, SAP continues to deliver innovations with SAP BusinessObjects 4.3 as part of a hybrid analytics configuration. This configuration allows SAP Analytics Cloud to access on-premise universes and Web Intelligence documents as a data source.

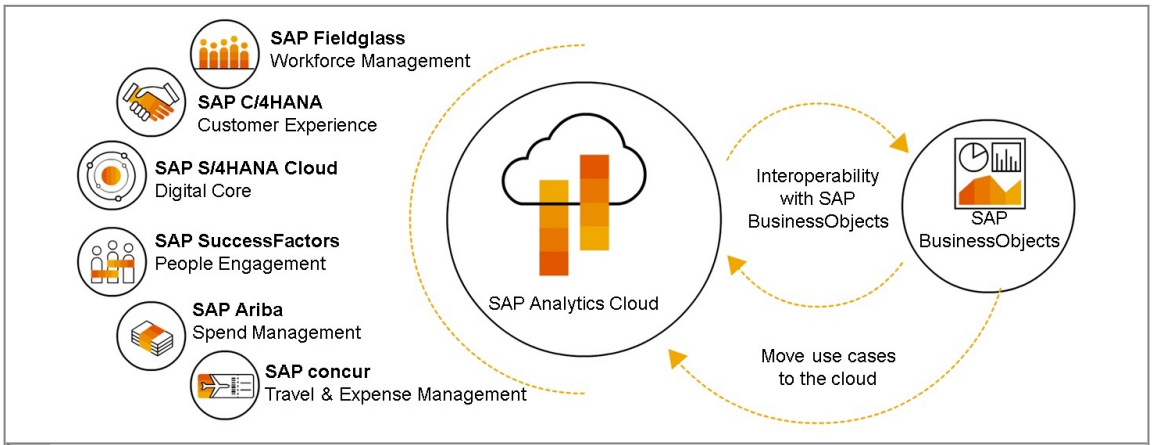


Figure 76: SAP BusinessObjects integration scenario

Identify data for SAP BusinessObjects models

SAP BOE Live Data Connect enables a live connection to on-premise data sources and allows customers to reuse existing data, queries, and security without moving the data to cloud. BI

4.3 allows customers to use both universes and Web Intelligence data models to build SAC Stories.

Live data connection can connect to a SAP Business Objects universe model and be used to create stories.

A **BusinessObjects universe** is an organized collection of metadata objects that enable business users to analyze and report on corporate data in a nontechnical language. These objects include dimensions, measures, and attributes.

The metadata object layer, called the business layer, is built on a relational database schema or an OLAP cube. The objects map directly to the database structures via SQL or MDX expressions.

A universe includes connections identifying the data sources so queries can be run on the data.

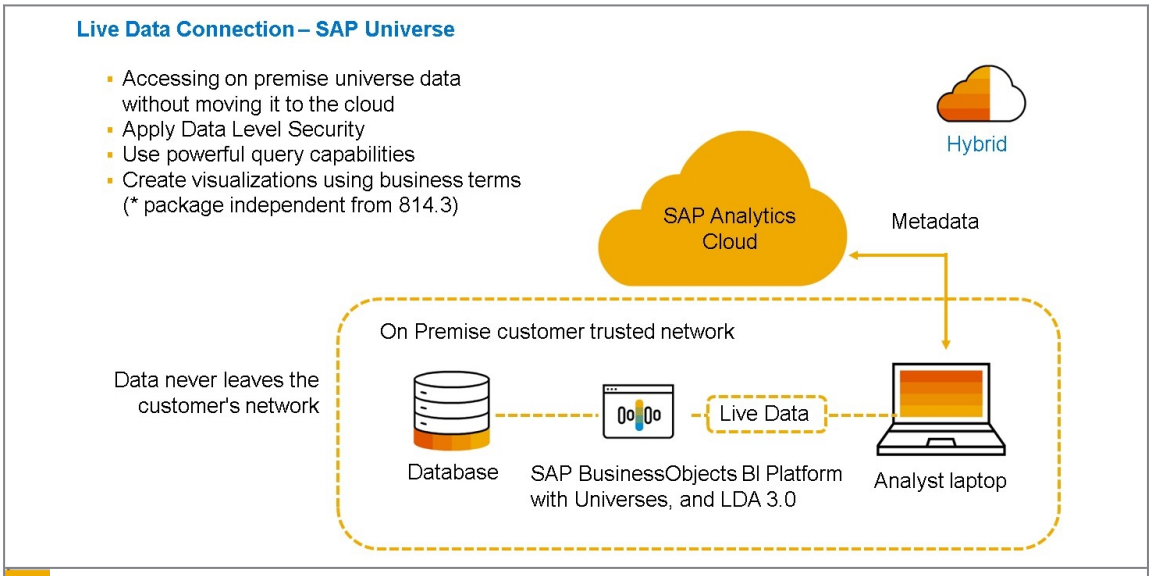


Figure 77: BI 4.3 SAP Universe

A **Web Intelligence** document is a report usually created from a query of a universe. It is the on premise equivalent of an SAP Analytics Cloud story. It can include calculations, called variables, that are created by the Web Intelligence formula editor and reside within the Web Intelligence document .wid file. This formula editor is very simple and intuitive yet also very powerful.

With a SAP Analytics Cloud live data connection, you can leverage these Web Intelligence documents, including any document variables (formulas) as data sources for SAP Analytics Cloud live models.

Note:
Web Intelligence *Variables* are not to be confused with SAP *BW Variables*. SAP BW Variables often render as prompts. Web Intelligence variables are simply calculations and formulas that have been defined with a name and will not prompt the user for any information.

LESSON SUMMARY

You should now be able to:

- Create a live model using SAP BusinessObjects data

Learning Assessment

1. The data analyzer can use import connections.

   Determine whether this statement is true or false.

   - ☐ True
   - ☐ False

2. When accessing data from SAP S/4HANA, the source object is a CDS view.

   Determine whether this statement is true or false.

   - ☐ True
   - ☐ False

3. What is the source when acquiring data from SAP HANA Cloud into SAP Analytics Cloud?

   Choose the correct answer.

   - ☐ **A** A core data services view
   - ☐ **B** A calculation view
   - ☐ **C** A composite view
   - ☐ **D** A responsive view

4. With a live data connection to SAP BW, data access is based on the BW source system's security.

   Determine whether this statement is true or false.

   - ☐ True
   - ☐ False

5. What is required to create an import data connection to an SAP BusinessObjects universe?

   Choose the correct answer.

   - [ ] A ODBC Connector
   - [ ] B SAP BTP Connectivity Cloud Connector
   - [ ] C JDBC Connector
   - [ ] D Generic Connector

Learning Assessment - Answers

1. The data analyzer can use import connections.

   Determine whether this statement is true or false.

   - [ ] True
   - [x] False

   Correct. The data analyzer can use live connections.

2. When accessing data from SAP S/4HANA, the source object is a CDS view.

   Determine whether this statement is true or false.

   - [x] True
   - [ ] False

   Correct. When accessing data from SAP S/4HANA, the source object is a CDS view.

3. What is the source when acquiring data from SAP HANA Cloud into SAP Analytics Cloud?

   Choose the correct answer.

   - [ ] **A** A core data services view
   - [x] **B** A calculation view
   - [ ] **C** A composite view
   - [ ] **D** A responsive view

   That's correct! Calculation views in SAP HANA Cloud are used to provide data to SAP Analytics Cloud.

4. With a live data connection to SAP BW, data access is based on the BW source system's security.

   Determine whether this statement is true or false.

   - [ ] True
   - [x] False

   Correct. With a live data connection to SAP BW, SAP Analytics Cloud uses existing SAP BW authorizations.

5. What is required to create an import data connection to an SAP BusinessObjects universe?

   Choose the correct answer.

   - [ ] A ODBC Connector
   - [x] B SAP BTP Connectivity Cloud Connector
   - [ ] C JDBC Connector
   - [ ] D Generic Connector

   Correct. The SAP BTP Connectivity Cloud Connector is required to create an import data connection to an SAP BusinessObjects universe.

UNIT 5 Working with geographic data

Lesson 1

UNIT OBJECTIVES

- Configure data for a geo map

Unit 5
Lesson 1

Configuring geographic data in a model

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Configure data for a geo map

Geo enrich your data

Geo enrich your data

Before you perform geospatial analysis in stories, you must first import coordinate data and enrich it in the modeler. This process creates a new column in the data view with an enriched format of latitude and longitude coordinates.

As a prerequisite, you must have a Microsoft Excel file .xslx or a .csv file with a location ID column that contains unique data, as well as latitude and longitude columns.

Geo maps are created in the modeler using location data. This acts as the foundation for the map in your story. There are two ways to geo enrich your data:

1. By coordinates: Geo enriching your data by coordinates enables you to use longitude and latitude coordinates. Your data may already include this information, but you could also enter it manually. These coordinates are used to create a map anywhere in the world.
2. By area name: During the creation of Geo dimension you can download all the currently supported countries, regions, and sub-regions. This option does not support an area enrichment with mixed granularity. For example, if enriching an area that has locations in both Country A and Country B, but Country A does not go down do the Sub-Region level, then the enrichment can only be created to the Region level at the lowest.

The figure *Geo Enrich Data by Coordinates* shows how you can enrich your data by coordinates.

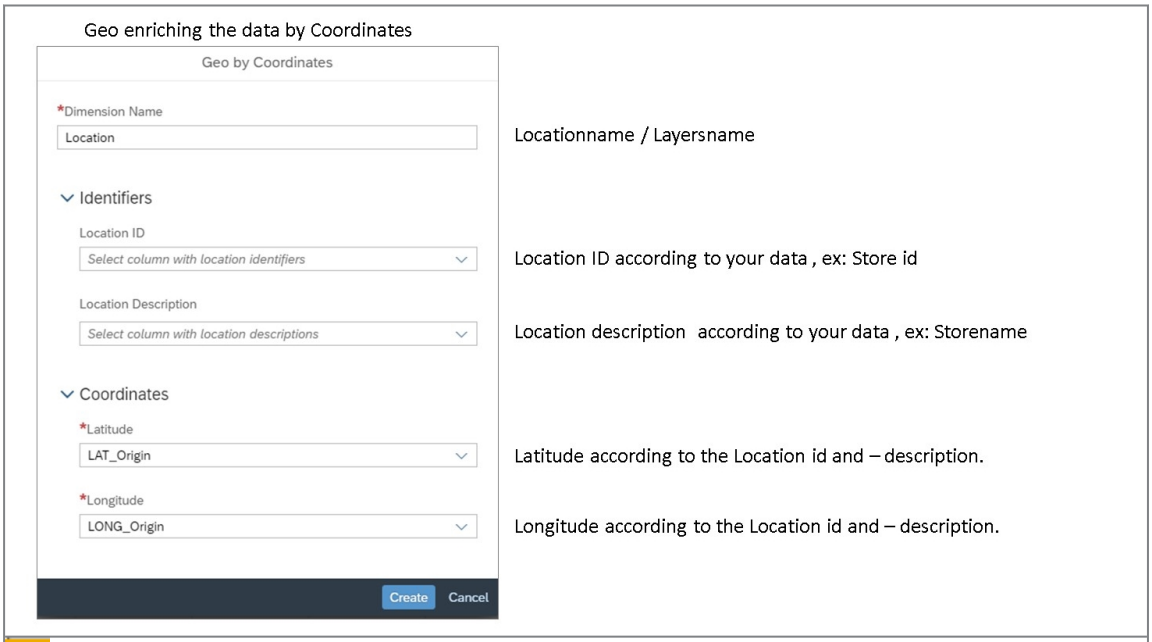


Figure 78: Geo enrich your data by coordinates

Points of interest

Points of interest are sets of geographical data that you can add to a geo map and analyze with reference to business data from a model.

You can add point of interest data from an Esri shapefile, a CSV or Excel file, or from an SAP HANA model with a geographical dimension.

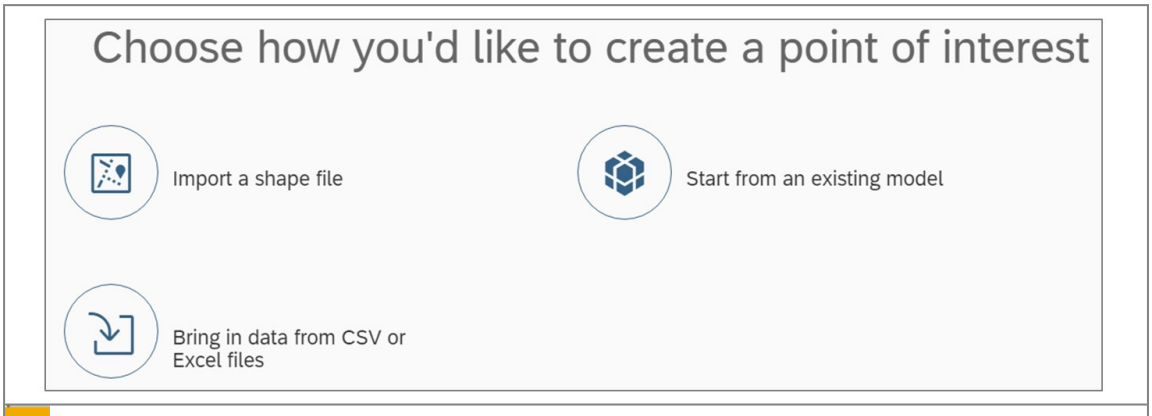


Figure 79: Point of interest

> Note:
> If you have a CSV or Excel file with clearly distinguishable names for latitude and longitude columns, you can simply drag and drop the file directly into your geo map.

SAP Analytics Cloud supports a finite number of spatial reference IDs. To avoid incorrectly displayed data or error messages, you should know the SRID used in your shapefile. Publicly available tools can help you recognize the shape identifier and thus determine the shapefile format.

You can view and maintain point of interest data in the Files list where you can filter on the Points of Interest file type to view the files. You can add or search points of interest, or select points of interest to delete.

You can set access permissions for point of interest data using the *Share* icon in the toolbar.

From the ESRI website you can access ESRI location data, such as Points of Interest: https://www.esri.com/en-us/arcgis/products/data/data-portfolio/places

LESSON SUMMARY

You should now be able to:

- Configure data for a geo map

Learning Assessment

1. When you geo-enrich data, the system concatenates the latitude and longitude columns.

   Determine whether this statement is true or false.

   - [ ] True
   - [ ] False

Learning Assessment - Answers

1. When you geo-enrich data, the system concatenates the latitude and longitude columns.

   Determine whether this statement is true or false.

   [X] True

   [ ] False

   Correct. When you geo-enrich data, the system concatenates the latitude and longitude columns.

UNIT 6 Defining data security

Lesson 1

UNIT OBJECTIVES

- Apply data access control

Unit 6
Lesson 1

Defining data access

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Apply data access control

Data access control in dimensions

SAP Analytics Cloud security in general

Security in SAP Analytics Cloud is used to control access to data and also access to objects. Controlling access to objects (i.e., who can create a model) is accomplished via roles. Controlling data access (i.e., who can view the data for what region) is accomplished primarily via data access control in dimensions.

Note:
Controlling access to data can also be carried out via roles.

SAP Analytics Cloud is delivered with several standard application roles. The roles you see will depend on the licenses included in your subscription. A role represents the main tasks that a user performs in SAP Analytics Cloud.

Roles are used mainly to control activities in SAC. In this context, roles are also object oriented; for example, user X can update dimension Y.

The Planner Reporter role:

- Includes all authorizations that are required to perform planning activities, such as revenue planning and automated discoveries. This role also grants authorizations for updating currency tables. This role is usually assigned to the user who does the planning and budgeting.
- This role also grants authorizations for viewing analytic applications and working with the data analyzer. It also grants authorizations for viewing custom widgets.

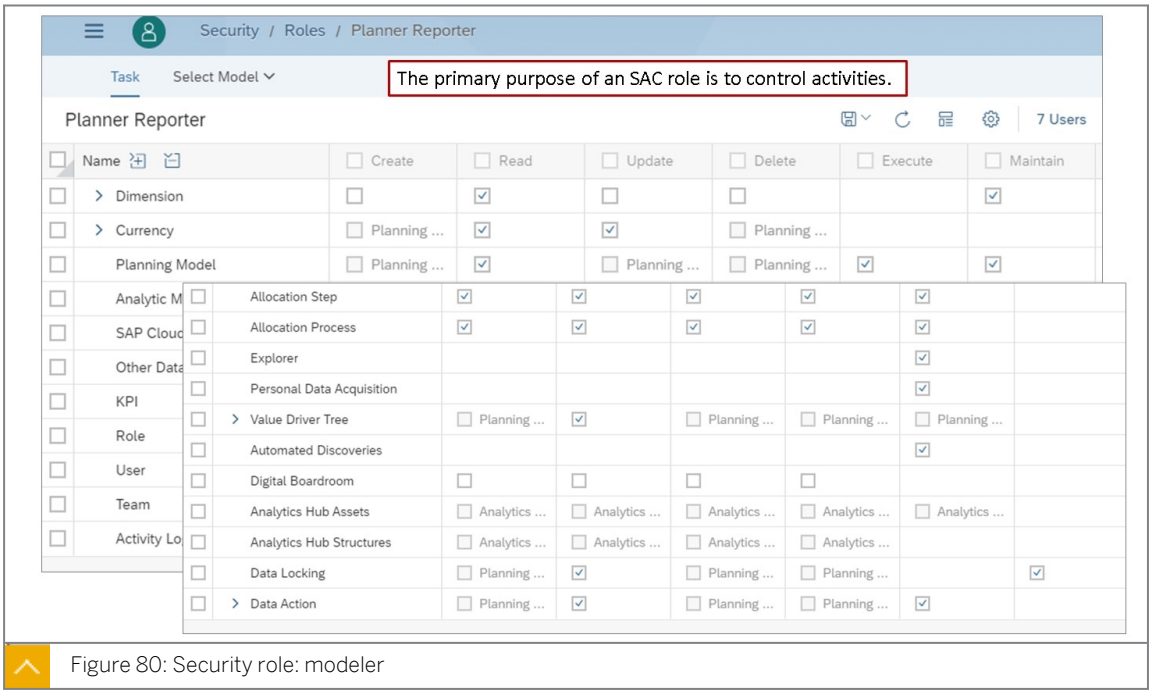


Figure 80: Security role: modeler

Basic permissions

- **Create:** Permits creating new objects of this item type. Users need this permission to create files and folders or upload data to an object, such as models, stories, point of interest, and others.
- **Read:** Permits opening and viewing an item and its content.
- **Update** Permits editing and updating existing items, including the structure of models and dimensions.
- **Delete:** Permits deletion of the item.
- **Execute:** Permits executing the item to run a process. For example, running a simulation using a legacy Value Driver Tree, or acquiring data from a data source.
- **Maintain:** Permits the maintenance of data values, for example adding records to a model, without allowing changes to the actual data structure.
- **Share:** Permits the sharing of the selected item type.
- **Manage:** This permission lets users manage content; for example, deleting content for any users, and resharing, copying, and moving content.

Visit SAP Help for additional information on permissions: https://help.sap.com/viewer/00f68c2e08b941f081002fd3691d86a7/release/en-US/93fec5646f144e109745ce74fd492c3f.html

Example of security permissions

Assignments are typically team-based with users assigned to teams and then roles assigned those teams. Roles are not typically assigned directly to users.

Permissions Available by Item

Name	Create	Read	Update	Delete	Execute	Maintain	Share	Manage	Notes
Dimension	X	X	X	X		X			Set the Maintain permission to permit adding members to a dimension without being able to change the actual definition. Set Update to allow changing the dimension definition itself.
Currency	X	X	X	X					
Planning Model	X	X	X	X	X	X			Set the Maintain permission to permit adding records of data to a model without being able to change the actual structure. Set Update to allow changing the model structure itself. Set Execute to enable planning features.

Figure 81: Example of security permissions

Model Preferences: Access and Privacy

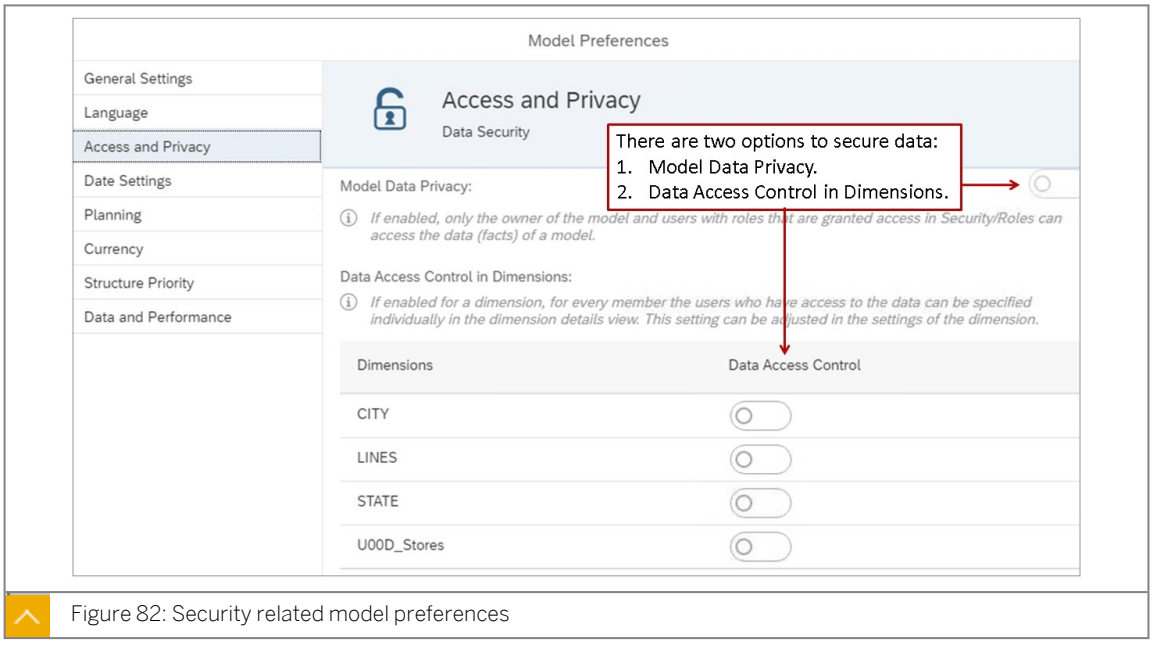


Figure 82: Security related model preferences

Data Access Control in Dimensions

You can restrict access to data in stories by setting read and write permissions for individual members. You can activate this security feature for any dimension in the model.

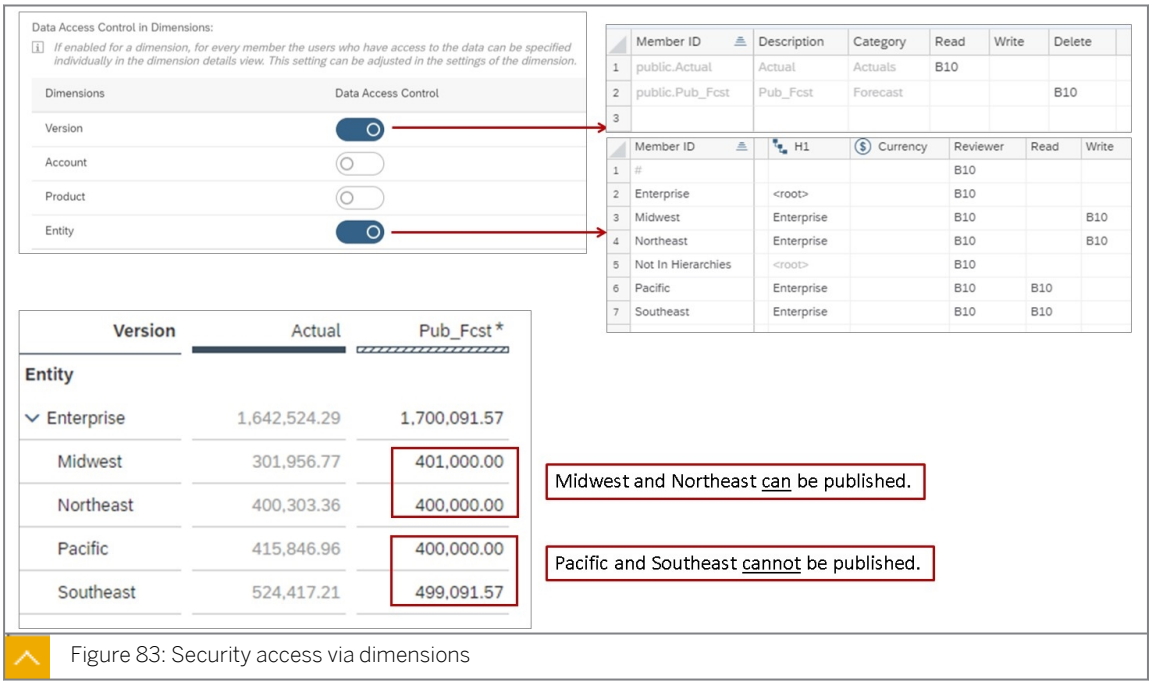


Figure 83: Security access via dimensions

Visit SAP Help for additional information on security access in dimensions: https://help.sap.com/viewer/00f68c2e08b941f081002fd3691d86a7/release/en-US/b46f2f74aea04c399ccb5c9b1bcc225e.html

You can enable data access restrictions using the *Data Access Control* (DAC) setting. When DAC is on, two more columns (*Read* and *Write*) are added to the dimension grid so that you can apply individual settings to each row. For the Version dimension, a *Delete* column is added as well as *Read* and *Write* columns to control which users can delete each public version.

When DAC is used with hierarchical data, you may want to switch on *Hide Parents*. Using this setting, you can restrict which dimension members can be seen in the Modeler. If this option is enabled, users will see only the members that they have at least read access to.

Each user who is granted write access for a member automatically receives permission to read the data as well. Likewise, a user who receives the delete permission for a member of the Version dimension also receives read and write permissions for it.

Adding version security to a model lets you restrict read, write, and delete access to public versions, to prevent other users or teams from changing them. Users who have read-only permission for public versions can still copy data to a private version that they can edit.

Users who do not have write permissions cannot publish into a public version. With delete permissions for a public version, a user can read, publish to, and delete a public version.

Implementing dimension-based security

In order to implement DAC / Dimension based security you must:

1. Develop a plan on who will have access to what data.
2. Activate *Data Access Control* for securing dimensions.
3. Maintain the user IDs in the dimensions.
4. Test the solution.

Note:

- Member IDs and hierarchies can be used.
- The dimensional assignments can be controlled by the business teams.

Model data privacy

Model data privacy

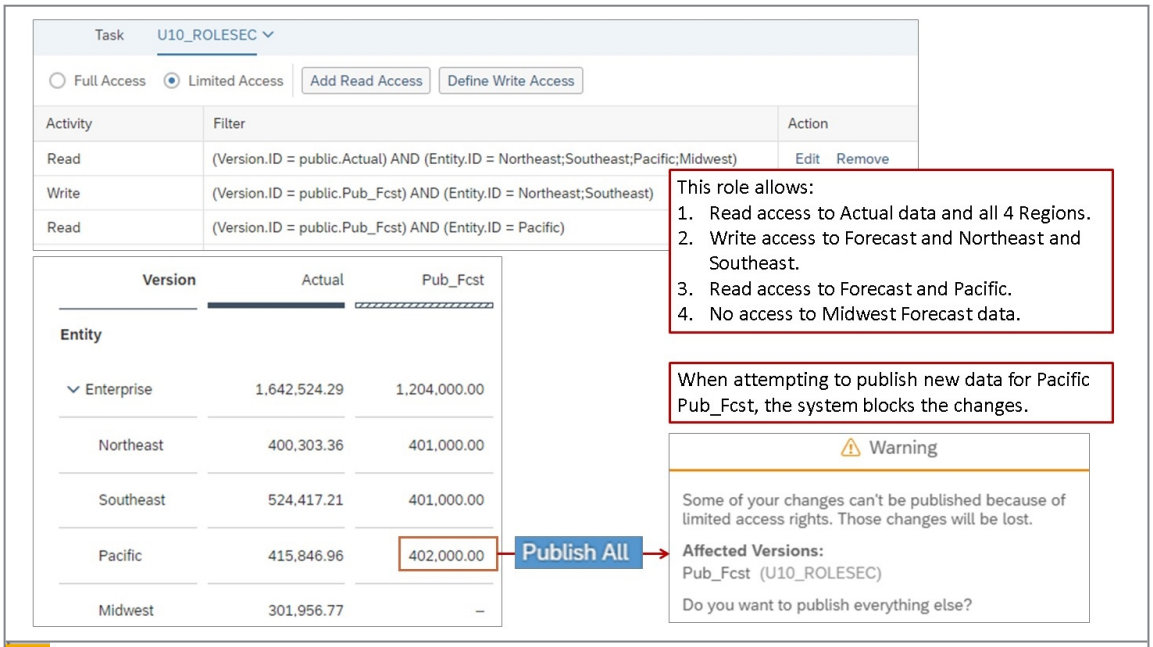


Figure 84: Using a role to control access to data

In order to implement *Role - based* data access:

1. Develop a plan on who will have access to what data.
2. Activate the *Model Data Privacy.*
3. Maintain the roles for data access.
4. Test the solution.

Note:

- Member IDs and properties can be used.
- Controlled mainly by the security teams.

LESSON SUMMARY

You should now be able to:

- Apply data access control

Learning Assessment

1. When data access control is enabled for an organization dimension, two properties are added.

   Determine whether this statement is true or false.

   - [ ] True
   - [ ] False

Learning Assessment - Answers

1. When data access control is enabled for an organization dimension, two properties are added.

   Determine whether this statement is true or false.

   [X] True

   [ ] False

   Correct. When data access control is enabled for an organization dimension, two properties are added: Read and Write.

UNIT 7 Optional: Applying your new skills

Lesson 1

UNIT OBJECTIVES

- Use the data modeling skills from this learning journey

Unit 7
Lesson 1

Practicing what you have learned

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Use the data modeling skills from this learning journey

LESSON SUMMARY
You should now be able to:

- Use the data modeling skills from this learning journey