

## Exercise 2 - Add Labels & Internationalization

In this exercise we enhance the data model such that we add

- language-independent labels for employees and companies
- language-dependent descriptions to products, product categories, regions & countries.

### Add Language-independent Labels to Employees & Companies

In this exercise we add labels for company names & employees. These obviously don't change with the user language and are thus **language-independent**.

#### Update Employee Labels

- Open view **4VD\_Employee**
- Update labels by clicking on the pencil  icon within the **Attributes** section of the **Model Properties**
  - Set **Semantic Type** of FULL NAME to Text
  - Set **Label Column** of EMPLOYEEID to FULL NAME

4VD_Employee					
Attributes (8)					Search 
Business Name	Technical Name	Data Type	Semantic Type	Label Column	Actions
<input checked="" type="checkbox"/> EMPLOYEEID	EMPLOYEEID	String(20)	None	FULL NAME	 
<input type="checkbox"/> NAME_FIRST	NAME_FIRST	String(20)	None		 
<input type="checkbox"/> NAME_LAST	NAME_LAST	String(20)	None		 
<input type="checkbox"/> SEX	SEX	String(1)	None		 
<input type="checkbox"/> PHONENUMBER	PHONENUMBER	String(30)	None		 
<input type="checkbox"/> EMAILADDRESS	EMAILADDRESS	String(50)	None		 
<input type="checkbox"/> MANAGERID	MANAGERID	String(20)	None		 
<input type="checkbox"/> FULL NAME	FULL_NAME	String(41)	Text		 

- **Deploy** your view

#### Update Company Labels

Repeat above steps for tying the company name to the company identifier

- Open view **4VD\_BusinessPartners**
  - Set **Semantic Type** of COMPANYNAME to Text
  - Set **Label Column** of PARTNERID to COMPANYNAME
- **Deploy** your view

## Update Analytic Model and Preview Results

The new modeling we have just done needs to now be loaded by the Analytic Model.

- Open **4AM\_SalesOrderItems** and refresh browser page
- Confirm that dimension PARTNERID now has a capital T (for Text) next to itself in the dimension list
- Confirm that RESPONSIBLE now also has a capital T next to itself in the dimension list
- **Deploy** your Analytic Model
- **Open Preview**
- Drill by PARTNERID and confirm that company name is now displayed
- Drill by RESPONSIBLE and confirm that employee's full name is now displayed
- Under **Rows** within the **Builder**, check drill-settings (three dots in dimension list on the right) and change presentation from ID and Description to **Description and ID**.

**⚠️** 2023-10-26: If Analytic Model hasn't picked up the metadata change, enforce an additional deploy by adding a blank space (" ") to the business name of the Analytic Model and hit deploy. Then repeat the steps above around the Analytic Model. This error is currently in fixing.

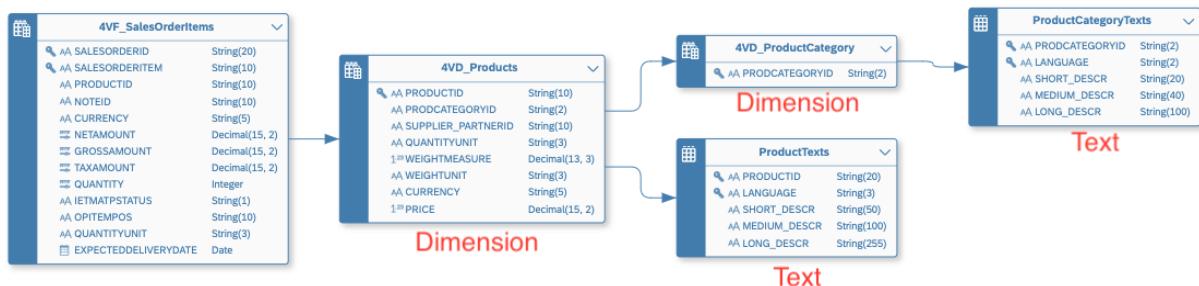
## Add Language-independent Labels For Products and Product Categories

Users feel more at ease if they see data in their native language rather than in some other language like English. To this end, SAP Datasphere provides the support of language-dependent labels. Depending on the data access language of the user (by default, this is their log-on language and native language), the labels for objects are drawn in that specific language.

### Modelling Guidance

The recommended modelling setup for language-dependent labels is as follows:

An entity of usage type Dimension uses a text association to link itself to an entity of type Text. The association uses the dimensions key field (not some other field!) to link to the key of the Text entity. If the key is comprised of several key fields (also known as "composite key" or "compound key"), then this is totally fine. The graphic below describes the relationship



In the example, dimension **4VD\_Products** has a text association to Text entity **ProductTexts**. The association uses the key **PRODUCTID** of the dimension in its mapping. In order to have also language-dependent labels for attribute **PRODCATEGORYID** (no key!) in the product dimension, there

should normally be an own dimension (here: 4VD\_ProductCategory) that is associated to the attribute and uses the respective field (PRODUCTCATEGORYID) in its mapping. This looks artificial in the example, but normally things like the product category are indeed dimensions in their own right, carry a multitude of attributes and thus merit an entity in its own right. This dimension 4VD\_ProductCategory in turn has a text association to a Text entity ProductCategoryTexts via the dimension's key field PRODUCTCATEGORYID.

It is not recommended (and a warning will be issued) when:

- the text association does not start from a dimension (but e.g. a Fact) or
  - the text association is on a non-key field of the dimension

Both cases are technically supported, but a warning will be issued. Both cases should rather have a dimension in the middle whose key is used in the text association (like for product category above). We'll see more of this below.

## Add Language-Specific Names to Product Dimension

- Open table **ProductTexts** and perform the following updates:
    - Change **Semantic Usage** to **Text**
    - Within **Attributes**:
      - Set **Semantic Type** of attribute LANGUAGE to Language
      - Set **semantic type** of attribute SHORT\_DESCR to Text
      - Set **Label Column** of attribute PRODUCTID to SHORT\_DESCR
  - **Deploy** the table
  - Open entity **4VD\_Products**
  - Within **Model Properties**, under **Associations**, add **Text Association** (+ sign) between 4VD\_Products.PRODUCTID and ProductTexts

- Ensure the mappings are correct (joined on PRODUCTID)

- **Deploy** your table

As you see, we follow the standard way for language-dependent texts to work: a dimension (4VD\_Products) with its key (here 4VD\_Products.PRODUCTID) uses a text association to map to a text entity (here ProductTexts) and its key (here ProductTexts.PRODUCTID)

### Add Language-specific Names to Product Category

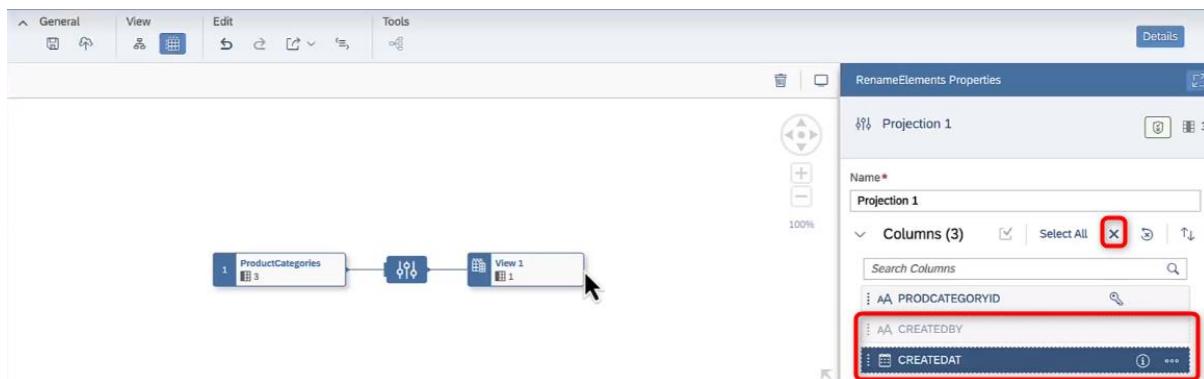
#### Create new Product Category Dimension

The product category so far is just an attribute of 4VD\_Products. In order to comply w above-mentioned modelling best practices, we require a real product category dimension that subsequently links out to the Text entity for product category texts.

We'll therefore create a new dimension view for product categories and base it on the imported table ProductCategories. We use the occasion to also hide fields that we don't want to expose to our analytics users (like CREATEDBY and CREATEDAT)

Since table ProductCategories only has three columns, the hiding of CREATEDBY and CREATEDAT leaves only a minimal dimension (consisting of just one field, PRODUCTCATEGORYID), but that's fine. In many more realistic cases there'd be own attributes to the category, like its category manager, a hierarchy on it or else, making it worthwhile to model this as a dimension in its own right.

- Create new **Graphical View** from the **Data Builder** home page
- Drag table **ProductCategories** into the canvas
- Add a projection node  and choose to exclude columns CREATEDBY and CREATEDAT by selecting the column name and hitting the X in the toolbar



- Choose final node (in this case, View 1) and set **Semantic Usage** to Dimension
- **Deploy** and save view as *4VD\_ProductCategory*

### Update Product Category Text Entity

- Open entity ProductCategoryTexts perform the following updates
  - Change **Semantic Usage** to Text
  - Within **Attributes**:
    - Set LANGUAGE to **Semantic Type** Language

- Set SHORT\_DESCR to **Semantic Type** Text
- Set SHORT\_DESCR as **Label Column** of PRODUCTCATEGORYID

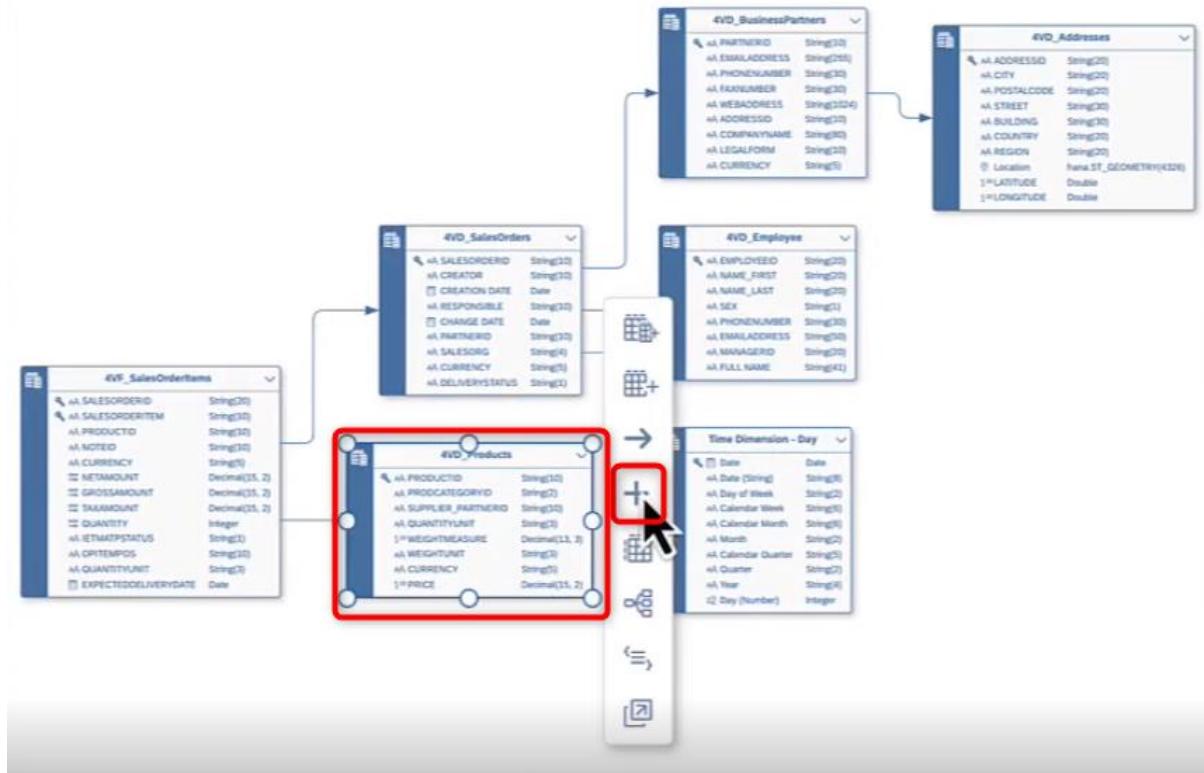
	Business Name	Technical Name	Data Type	Semantic Type	Label Column	Default Value	Not Null
<input type="checkbox"/>	PRODCATEGORYID	PRODCATEGORYID	String(2)	None	SHORT_DESCR		<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	LANGUAGE	LANGUAGE	String(2)	Language			<input checked="" type="checkbox"/>
<input type="checkbox"/>	SHORT_DESCR	SHORT_DESCR	String(20)	Text			<input type="checkbox"/>
<input type="checkbox"/>	MEDIUM_DESCR	MEDIUM_DESCR	String(40)	None		Enter a string	<input type="checkbox"/>
<input type="checkbox"/>	LONG_DESCR	LONG_DESCR	String(100)	None		Enter a string	<input type="checkbox"/>

- **Deploy** your table
- Open 4VD\_Products and add an **Association** between 4VD\_Products and 4VD\_ProductCategory
- Ensure mapping is correct (mapped on PRODUCTCATEGORYID)
- **Deploy** your view
- Add a **text association** on view **4VD\_ProductCategory** by opening the view and adding a text association to ProductCategoryTexts that maps 4VD\_ProductCategory.PRODUCTCATEGORYID to ProductCategoryTexts.PRODUCTCATEGORYID.
- **Deploy** 4VD\_ProductCategory

### Update ER Model

We should update the ER model with the new objects and their relationships in order to always have a good overview of our overarching model. This is easy to do since we can leverage the newly drawn associations to add the respective entities.

- Open ER Model **4EM\_Overview\_Simple**
- Select node 4VD\_Products and choose plus sign (+)



- Choose to add related entities **4VD\_ProductCategories** and **ProductTexts**
- Choose to add related entities of **4VD\_ProductCategories** - again via the plus sign.
- **Deploy** your ER model

### Update Analytic Model and Preview Results

The new metadata needs to be considered also by the Analytic Model. For it to take note of the updated metadata, the Analytic Model page needs to be loaded newly or refreshed. Subsequently you should save and deploy.

- Open **4AM\_SalesOrderItems** and refresh browser page
- Confirm that dimension **PRODUCTID** now has a capital T (for Text) next to itself in the dimension list
- Open node **PRODUCTID** and choose to add associated dimension **PRODCATEGORYID**. Note that it also has a capital T (for Text)

#### Associated Dimensions (1)



- **Deploy** your analytic model

- Open **Preview**
- Drill by PRODUCTID and confirm that Product Names are now being displayed
- Drill by PRODCATEGORYID and confirm that the category ID name is now displayed
- Change settings of your user by clicking on your user profile in the top-right of the application. In section **Language & Region**, change **Data Access Language** from English to French.

The screenshot shows the SAP Fiori Settings interface. On the left is a sidebar with various settings categories: User Account, Appearance, Home Screen, Language & Region, UI Settings, Privacy, and Authorized Consent Settings. The Language & Region category is selected and highlighted in blue. The main area is titled "Language & Region". It contains several dropdown menus and buttons:

- Language:** Set to English.
- Data Access Language:** A dropdown menu showing "English" (selected) and "Français". The "Français" option is highlighted with a red rectangle and has a small hand cursor icon over it.
- Date Formatting:** Български (Bulgarian).
- Time Formatting:** Català (Catalan).
- Number Formatting:** Čeština (Czech).
- Scale Formatting:** Cymraeg (Welsh).
- Currency Position:** Dansk (Danish).
- Other Options:** Deutsch (German), Ελληνικά (Greek), English (United Kingdom), Español (Spanish), Español (México), Eesti (Estonian), فارسی (Persian), Suomi (Finnish), and another "Français" option at the bottom.
- Buttons:** A "Close" button in the bottom right corner.

- Confirm with **Save**

- Repeat drilling by PRODUCTID and PRODCATEGORYID. Confirm that you now see French texts for products and their category.

PRODCATEGORYID		PRODUCTID		Mesures	NETAMOUNT
	↑↓		↑↓		NETAMOUNT ↑↓
BX	VTT	BX-1011	VTT Vintage 1011	903,372.00 \$US	
		BX-1012	VTT Saut 1012	1,376,550.00 \$US	
		BX-1013	VTT Saut Luxe I	1,629,870.00 \$US	
		BX-1014	VTT Saut Luxe II	2,834,852.00 \$US	
		BX-1015	VTT Optimal	1,088,061.00 \$US	
		BX-1016	VTT Optimal II	1,103,102.00 \$US	
CB	Croiseur	CB-1161	La Plage	1,405,278.00 \$US	
		CB-1162	La Plage Limité	1,432,400.00 \$US	
		CB-1163	La Plage Or	1,005,984.00 \$US	
CC	Vélo de cyclo-cross	CC-1021	Cyclone Base	4,092,088.00 \$US	
		CC-1022	Cyclone Vitesse	4,171,200.00 \$US	
		CC-1023	Cyclone III	5,016,646.00 \$US	
DB	Vélo de descente	DB-1081	Racine Base	5,116,087.00 \$US	
		DB-1082	Capricorne	4,426,250.00 \$US	
		DB-1083	Capricorne II	4,298,415.00 \$US	
EB	Vélo Électrique	EB-1131	Le Flash	33,106,500.00 \$US	
		EB-1132	Le Flash II	41,634,700.00 \$US	

⚠️ 2023-10-26: If Analytic Model hasn't picked up the metadata change, enforce an additional deploy by adding a blank space (" ") to the business name of the Analytic Model and hit deploy. Then repeat the steps above around the Analytic Model. This error is currently in fixing.

### Add Language-independent Labels For Countries & Regions

For completeness sake, we should quickly add labels to countries & regions as well - "Germany" is just easier to read than "DE", isn't it? With what we have learned so far, that's just a couple of minutes more work, but it'll make life quite a bit simpler for our users.

For simplicity, we do not go via own dimensions for countries and regions, but associate the text entities directly to the address dimension. This will give us warnings, but we can always clean this up later.

### Add Country Text

- Open entity **Countries** and perform the following updates:
  - Add **Semantic Usage** to text
  - Within **Attributes**:
    - Set LANGUAGE to **semantic type** Language
    - Set COUNTRYTEXT to **semantic type** Text
    - Set COUNTRYTEXT as **label column** of COUNTRYCODE
- **Deploy** your table

### Add Regions Text

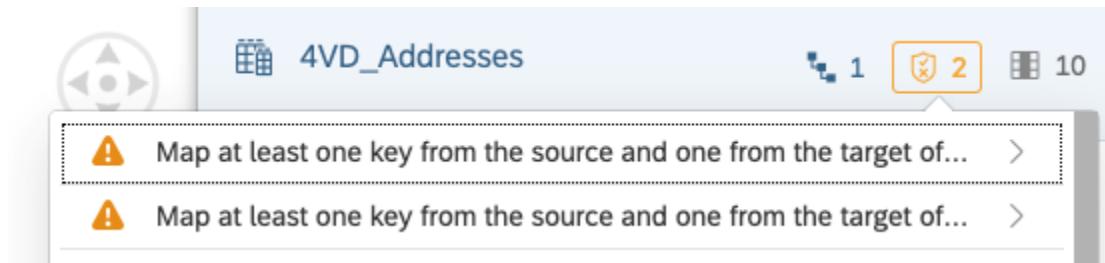
- Open entity **Region** and perform the following updates

- Add **Semantic Type** to Text
- Set LANGUAGE to **Semantic Type** Language
- Set REGIONTEXT to **Semantic Type** Text
- Set REGIONTEXT as **Label Column** of REGIONCODE
- **Deploy** your table

### Update Addresses view

- Open entity **4VD\_Addresses** and refresh browser page
- Add **Text Association** between **4VD\_Addresses** and its attribute COUNTRY and Countries.COUNTRYCODE
- If mapping is missing, simply drag COUNTRY column to COUNTRYCODE
- Add **Text Association** between **4VD\_Addresses** and its attribute REGION and REGION.REGIONCODE
- If mapping is missing, simply drag REGION column to REGIONCODE

**⚠** You'll get a warning about key mapping in the properties of **4VD\_Addresses**. This is because of the above-mentioned [modelling best practices](#). Here, the text association from **4VD\_Address** to the text entities for country text & region text are not using the key of **4VD\_Addresses**. For cleaner modelling, there should be a Country dimension and a Region dimension that are put in between like we did above for product categories.



- **Deploy** your view. In the warning popup, choose **Deploy anyway**.

### Update ER Model

We should update the ER model with the new objects and their relationships in order to always have a good overview of our overarching model. This is easy to do since we can leverage the newly drawn associations to add the respective entities.

- Open ER Model **4EM\_Overview\_Simple**
- Select node **4VD\_Addresses** in the canvas and choose plus sign (+)
- Choose to add related text entities **Countries & Regions**
- **Deploy** your ER Model

### Update Analytic Model and Preview Results

The new metadata needs to be considered also by the Analytic Model. For the metadata to persist, the Analytic Model page needs to be refreshed. Subsequently you should save and deploy.

- Open *4AM\_SalesOrderItems* and refresh browser page
- Confirm that dimension COUNTRY now has a capital T next to itself in the dimension list
- Confirm that REGION now also has a capital T next to itself in the dimension list
- **Deploy** your Analytic Model
- Open **Preview**
- Drill by COUNTRY and confirm that country name is now displayed in French
- Drill by REGION and confirm that region name is now displayed in French
- Change settings of your user by clicking on your user profile in the top-right of the application. In section **Language & Region**, change **Data Access Language** from French to English
- Repeat drilling by COUNTRY and REGION and confirm that all their texts are now displayed in English again.

⚠ 2023-10-26: If Analytic Model hasn't picked up the metadata change, enforce an additional deploy by adding a blank space (" ") to the business name of the Analytic Model and hit deploy. Then repeat the steps above around the Analytic Model. This error is currently in fixing.

## Summary

Great work! You were able to enhance the data model by adding language-independent and language-dependent labels, expanded our ER model and tested our improved model directly in the Analytic Model data preview.

Continue to - [Exercise 3 – Hierarchies](#)