

The Tool

Disclaimer

This prototype (the file) is made available as open source.

The terms

This file is created in MsAccess and will be provided “as is” and without any form of warranty, including without limitation merchantability, fitness for a particular purpose, absence of defects or errors, accuracy, non-infringement of intellectual property rights. In no event shall TenneT nor its employees be liable for any direct, indirect, incidental, special, exemplary or consequential damages (including, but not limited to, procurement of substitute goods or services, loss of use, data or profits, or business interruption) however caused and on any theory of liability, whether in contract, strict liability, or tort (including negligence or otherwise) arising in any way out of the use of this MsAccess file, even if advised of the possibility of such damage. The file is not to be used for whatever production environment.

This MsAccess file does not include a license to use the MsAccess program of Microsoft, nor does it grant permission to use the trade names, trademarks, service marks, or names of TenneT and/or third parties. The user of this MsAccess file is solely responsible for obtaining the applicable licenses from Microsoft or any other license necessary.

If you make modifications to the MsAccess file, please provide the modification including explanation to the author. The author will continuously improve the file and make all updates available.

If you use the principle / tool / approach as explained in your software and/or in a publication/article/expression (commercially or not commercially), please include the credentials of the author (TenneT / Coen Hendrikx).

Versions

If you want to stay informed about the development of this prototype and receive all updates, please like the LinkedIn article. You will then be notified about relevant news, and the latest version of the tool will be continuously published on the page.

The article has been posted on my [LinkedIn page](#).

EAV; the main advantage is also a disadvantage

The model utilized in our prototype is based on the Entity Attribute Value (EAV) approach, renowned for its flexibility. This flexibility is both an advantage and a disadvantage.

Please read the Wikipedia for details concerning EAV: [Entity-attribute-value model - Wikipedia](#).

The advantage lies in the ability to store a wide range of data types. When registering a record (instance) belonging to a specific class, you can define the class (Entity) itself (which would correspond to a main table in a relational database) and its attributes (equivalent to columns in-, or referencetable to a main table). Moreover, you can establish relationships between objects within the same or different classes (resembling relationships in a relational database). All of this can be achieved through configuration.

However, this flexibility also presents a challenge. As there are minimal constraints imposed by the tooling, the design of the data model heavily relies on the discipline of the data custodian or data steward. It is crucial for the responsible implementer to exercise discipline and ensure that all data adheres to the required data quality standards.

The instruction

The tool facilitates the process of setting up the relational data model and registering the data. The first part (setting up the data model) involves several steps. The initial step is to define entity classes, which represent the types of instances that you want to register. In the context of the electricity industry, entity classes could include connection points, generation units, allocation points, or areas.

For the purpose of our demonstration, we will take a detour and focus on implementing the following classes: Road, Place, Road type and Material. These classes will serve as examples to showcase the functionality of the tool and illustrate how instances belonging to different entities (entity classes) can be registered and managed within the system.


Next, you will proceed with registering the specific information you want to track about the instances. This can be seen as “the set of questions you want to ask about a particular instance or record pertaining to a specific entity class”. For example, you may want to register a name for a road or place.

These questions, “attributes”, are not only registered independently but also linked to one or more entities. This ensures that specific questions (attributes) can only be asked to entity instances that belong to the relevant entity class. For instance, you want to be able to register a license plate for a car, but not for a road. By associating attributes with specific entity classes, you can prevent the registration of irrelevant information and maintain data consistency.

The next step is to create relationship types. A relationship establishes a connection between two instances (records). In our example, we will connect a road to a place, so we need to register the relationshiptype “road connects place”. We will also connect the appropriate entity classes to the relationshiptype.


The data model is designed to handle various types of relationships (one-to-many, many-to-one, and many-to-many).

The above, step by step


Note: When you see this icon , it means that the explanation "starts" from the start form (splash screen). This splash screen shows the same icon.

Open the file


- Obviously, you need to have MsAccess installed on your computer.
- Make sure
 - o You are working on a big enough screen (laptop screens may not display the whole screen)
 - o you have the application (the file) stored on a local drive (C:\, D:\, ..) (it will work on a network environment, but slower.)
- Simply double click the file.

If the splash screen  does not appear, please double-click on the left side to open the Form "frm 0 Start"¹.

Defining the model: Entity (Class)

- : Click **Entity Class**
- Select an empty record.

The bottom record in the list is the empty record; it contains only empty fields.
(If you define your first entity class, you will only see one empty record)
- Register:
 - o ClassName: Road
 - o ClassCode: Road
 - o Choiceproperty = (don't tick the button)
 - o Hyperlink: Leave blanc (for now)
 - o Icon: If you have an icon, add it:
 - R-mousebutton: select Manage attachments, Click add, select the icon, click "Open" and confirm by clicking "Ok".




The icons used in this document, as well as numerous other icons, can be found at <https://www.iconarchive.com/>. If you intend to use these icons, please ensure that you comply with the licensing information provided on the website itself.
- Repeat the registerstep to register the following class:
 - o Place
- Close the window
- : Press F5 to refresh data
 - o If correct, the Entityclass window is filled with the added entityclasses.

That was the definition of entityclasses. Entityclasses can be seen as the main tables in a relational database.

¹ The splash screen should automatically launch after you have opened the file by double-clicking it. If that is not the case, the correct setting is missing in the options. Go to File - Options - Current Database, and under 'Application Options', select 'Display Form' and set it to 'frm 0 start'.


Defining the model: Atributes

Next step: the attributes. Or, “the questions you would like to ask concerning the instances”.

- : Click **Attributes**
- Register the attributes (“questions”) you would like to register.
- Before registering the attribute, make sure the applicable unit is available. You can see all available units (Currently: Number, Text, Hyperlink) via the select-list Datatype.
 - o If the required unit is not available, it must be registered first. The predefined datatypes (txt, num, hyperlink) are not visible. All other datatypes have to be registered.
 - o You can register a new datatype via:
 - 
 - Click **Units/Datatypes**
 - Go to the empty line (New)
 - Register the datatypecode and the datatype name and (optionally) the sign (€, kWh, mm, etc.) that must be shown where a value is registered².
 - Close the subwindow.
 - o Register 2 new datatypes by following the aforementioned steps:
 - **Datatype Code / Datatype Name / Sign:**
Datatype 1: Kilometre / Kilometre / km
Datatype 2: Meter / Meter / mtr
- Back to registering the attribute:
 - o : Click **Attributes**
 - o In the attributewindow, go to the first empty line starting with (New)
 - o Define the attribute (“the question”).
 - o Sortingorder:
 - You can optionally define the sorting order.
 - Lowest number will be on top
(sorting is done in alphabetical (not numeric) order)
 - You can leave this empty for now.
 - o Overlap allowed:
 - Do not click “Is overlap allowed”.
This function will be explained later
 - o Select the applicable datatype.
 - o Hyperlink: leave empty for now. This will be explained later.

² Register the sign if applicable. This sign will be shown after the recorded value. So a currency value like € will be shown in a form like: <value> <sign>: 739 €. The sign field can be left empty. If no value is recorded, no sign will be shown.

- The following attributes can be defined:
(just register the attribute and select applicable datatype)
 - o **Name** (datatype is text)
 - o **Shortname** (datatype is text)
 - o **Length** (datatype is kilometre)
 - o **Width** (datatype is metre)
 - o **Number of inhabitants** (datatype is number)



- Go back to Start form / Home menu  (close the attributes window)
- Press F5 to refresh the screen.

That's it. We have defined questions (attributes) and entity-classes. Classes can be compared to main-tables in the relational database equivalence, while attributes to some extent represent the columns in those tables or reference tables with a relationship to the main tables.

Defining the model: Relate attributes to classes

To ensure that we cannot accidentally ask inappropriate or irrelevant questions, we need to enforce the constraint of asking only the right kind of questions for each entity-class. For example, asking about the number of inhabitants of the freeway between the cities of Nijmegen and Arnhem would not make sense, as freeways do not have inhabitants. On the other hand, asking about the number of inhabitants of the city of Nijmegen would be a relevant question.


By defining the allowed attributes per entity class, we can restrict the types of questions that can be asked, ensuring that only appropriate and meaningful inquiries can be made. This helps maintain data integrity and prevents inconsistencies in the system. Or, in the relational equivalence; make sure to have the correct columns defined per table.

- : Click on “Entity Class”
- Click on “Settings” for the entity class **Road**.
The entity class detail screen (“Entity Class Attributes Settings”) will open up.
- On the left side of the screen, you will find the already registered info (name, code and if registered, an icon)
- On the right side of the screen, there will be a subwindow where you can select attributes.
- For the class “Road,” select the applicable questions (attributes) by doing the following: (while registering, leaf the validity period and privacy box for now)
 - o Select *Shortname*, and choose “1st element” as the identifying attribute
 - o Select *Name*, and choose “2nd element” as the identifying attribute
 - o Select *Length*, and leave the Identifying attribute dropdown option empty.
- Close the window
- Next, click on “Settings” for the record **Place**
- We are going to relate the applicable questions for the class “Place”
 - o Select *Name* and choose “1st element” as the identifying attribute.
 - o Select *Shortname* and choose “2nd element” as the identifying attribute.
 - o Select *Number of inhabitants* and choose “3rd element” as the identifying attribute.
- Go back to the Splash screen / Home menu , by closing the windows by clicking “Close.”

By completing these steps, we have effectively registered the EAV (Entity-Attribute-Value) variant of defining tables and columns within those tables.

Registering masterdata: The instances


Now, we are going to register our initial set of master data.

- : Click on “Add Instance”
- Select the relevant entity class. For this example, let’s define a city, so choose “Place.”
- Next, click on the drop-down list in the subwindow and take a look at the available options.
 - o You will only find the attributes (questions) that you have related to the class “Place”
- Select Shortname, and register “NM” as the value
 - o You can specify the validity period when entering an attribute value. The "start date" is filled in with today's date by default. The "end date" field is empty. If you don't yet know when a value is valid until, or if the validity end date is not applicable ("forever"), you leave this field empty. For this example, leave both date fields untouched to specify the attribute shortname that is valid as of today.
- New line, select Name, and register “Nijmegen”
 - o The attribute “Name” was defined as the first identifying attribute for the entity Place. This is the key-value for all recorded instances. Every instance (record) from the entity class Place should have registered a 1st identifying attribute. Even though it is possible to register an instance without a 1st identifying attribute-value, for the sake of consistency it is recommended to register consistently.
- New line, select Number of inhabitants, and register “170000” as the value
- Close the window.


Congratulations! You have successfully registered your first master data record.

Now, let’s repeat the same process for Arnhem (AH, Arnhem, 150000).

Let’s have a look at our entry.


- : Click in the list “Entity Class” on “Place”
- Click on “Entity Class Instances”
Here, you can see all the records related to the “Place” class that are currently present in the database (only 2 records at the moment). The column headers in this list represent the attributes we defined as the 1st, 2nd, and 3rd attributes for the Place class; Name / Shortname / Number of inhabitants.
- Let’s open the details of Nijmegen by pressing the “Details”-button
- You can view the details you registered.
 - o At the top of the form (header), you will see the Name (1st element), Shortname (2nd element), and the number of inhabitants (3rd element) in the order you defined.
 - o In the body section,
 - you will find in the tab “Text attributes”
 - the Shortname : NM
 - the Name: Nijmegen
 - you will find in the tab “Quantity Attributes”
 - the number of inhabitants: 170000
 - you will find all the registered attributes in the tab “All attributes (incl. History)”
- Close this window, and also take a look at the details of Arnhem.


Let’s continue registering masterdata, by defining some roads.

- : Click on “Add Instance”, and select the class “Road”
- Choose the attribute “Shortname”, and enter “A325” (which is the code for the freeway between Arnhem and Nijmegen)
- Select the attribute “Length” and enter “28.”
 - o Please note that you won’t be able to select a unit (e.g., kilometer) as you have already preselected this unit in the attribute section.
 - o Only enter the number 28. Do not write “km” after the number, the predefined unit will be shown in the application.
- Since the freeway doesn’t have a specific name, you do not need to register anything in the “Name” attribute
- Close the popup screen

Next, let’s add another road:



- Click on “Add Instance” again and select the class “Road”
- Choose the attribute “Name” and enter “Energyroad”
- Choose the attribute “Shortname” and enter “Energyroad” again
- Select the attribute “Length” and enter “3”
- Close the popup screen

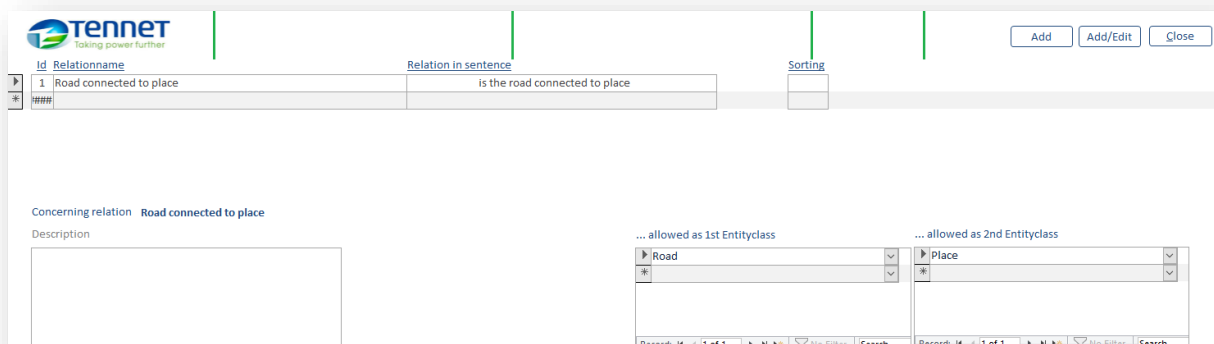
Have a look at all the roads: Via Splash screen / Home menu , select Road, and click “Entity Class Instances”.

Check the details if you want. When you are finished, close all the forms until you are back at the Splash screen / Home menu .

Defining the model: Relationtype (Main to main)

We proceed by connecting the roads to the cities.

- : Click on “Relation Class”
- Click on the “Add/Edit” button to create a new relationclass record
- Go to the first empty line, and enter the following details for the relation class
 - o Relationname: “Road Connected to Place”
 - o Relation in sentence: “is the road connected to”
 - o Optionally, define the relationtype (1-to-many, many-to-1, many to many) (There is no functionality linked to this input. For information purposes only.)
 - o Sorting: Optional: Register a number by which the records should be sorted. (1 being the most important, up to 9)
 - o Make sure both “Overlap allowed” booleans are ticked: (The underlying functionality will be explained later)
 - o Optionally, you can provide a description for the relation class and/or description for validation info at the bottom of the screen.
 - o Pay attention to the two sub windows at the bottom right:
 - We want to establish a relationship called “Road Connected to Place”
 - Following the title, we only want to be able to select
 - o Instances from the entityclass “Road” as the 1st component
 - o Instances from the entityclass “Place” as the 2nd component
 - Left sub window (... allowed as 1st entityclass) select class “Road”
 - Right sub window (... allowed as 2nd entityclass) select class “Place”
 - o Close the window and navigate back to 
 - o Press F5 to refresh the splash screen



The screenshot shows the 'Relation Class' configuration interface. At the top, there are buttons for 'Add', 'Add/Edit', and 'Close'. Below these are three main input fields: 'Relationname' (containing 'Road connected to place'), 'Relation in sentence' (containing 'is the road connected to place'), and 'Sorting' (empty). Below the main form, there are two sub-windows for selecting entity classes. The left sub-window is titled '... allowed as 1st Entityclass' and has 'Road' selected. The right sub-window is titled '... allowed as 2nd Entityclass' and has 'Place' selected. At the bottom left, there is a section for 'Concerning relation Road connected to place' with a 'Description' field. At the bottom right, there are two record lists, each showing 'Record: 1 of 1' and 'No Filter'.



Registering a relation between two instances

Now that we have defined the relation class, it's time to register a relation record for the relation between a place and a road.


The A325 freeway is a road between the cities Arnhem and Nijmegen.

We will (always) register the relation from the perspective of the 1st instance (record).

Here are the steps:


- : Open the class "Road" by selecting the entity class "Road" and clicking on "Entity Class Instances."
- Select the A325 road and click on "Details".
- Click on "Add Relation (select 2nd)".
- Choose the applicable relation class (currently only one listed): "Road connected to place"
- In the "Select 2nd Entity" field, choose the place "Nijmegen"
 - o Note that you can only select a place because of the setting in the relation class, where the 1st component must be from the class "Road" and the 2nd component must be from the class "Place"
- Close the subwindow (Close and Save). You will be back in the details screen, which displays the details of the A325 road.
- From this detailscreen , click on "Add Relation" again.
- Repeat the actions, now relate the road A325 to the place "Arnhem".
- Return to the Spash screen / Home Menu .



The Energy road is a road within Nijmegen. To connect the Energy road to Nijmegen, follow these steps:

- : Click on "Road" and then select "Entity Class Instances". This will display a form showing all the recorded roads
- Locate the Energy road and click on "Details" to view its specific details
- Click on "Add Relation (select 2nd)" to establish a connection
- Select the relationtype "Road Connected to Place" from the available options
- Choose "Nijmegen" as the place to which the Energy Road is connected
- Click "Close and Save"

In the details screen, tab "Relations", you can see the registered relationships.



Defining the model: Reference Entity (Choice Entity Class)

- Roads are constructed using specific materials. Let's define "Material Type" as an entity class by following these steps, similar to registering the "Place" class. However, this time we will select the "Choice Property" option.
 - : Click on "Entity Class"
 - Select an empty line
 - Register a new Entity Class called "Material type"
 - Class Name: Material type,
 - Class Code: Material,
 - Choiceproperty = (Tick the checkbox to enable this option)
 - If desired: add an icon
 - Click on "Settings",
select "Shortname" as attribute and
register 1st element as identifying attribute
 - Close the window, to return to the Entity Classes form.

- We have various types of roads, such as freeways or local roads. Let's define "Road Type" as an entity class:
 - If you are at , click on "Add / Edit Entity Class"
 - Register a new entity class called "Road Type"
 - Name: Road Type
 - Code: Road Type
 - Choiceproperty = (tick the checkbox to enable this option)
 - If desired: add an icon
 - Click on "Settings",
 - Select "Shortname" and choose 1st element
 - Select "Name" and choose 2nd element
 - Close the windows and return to Splash Screen / Home Menu .
 - Press F5 to refresh the screen


Registering referencedata: The instances.

To register the instances follow these steps:



- : Click on “Add Instance” to begin registering the instances.
- For road types: Register 3 instances (records), every single instance representing a road type:
Name (Shortname):
 - Freeway (FW),
 - Main road (MR),
 - Local road (LR).
- For Material types: Register 4 instances, every single instance representing a materialtype:
 - Concrete,
 - Asphalt,
 - Gravel,
 - Bituminous
- Close the window to complete the registration of the instances
- Return to the Splash screen / Home Menu .

Defining the model: Relationtype (Main to reference)



To be able to relate a road (main class) to a material type (reference class), follow these steps:

- : Click on “Relation Class”
- Click on “Add/Edit”
- Click on the empty line under “Relation Name” in the screen
 - o Relationname: “Road is made of material”
 - o Relation in sentence: “is the road fabricated with the material”
- Overlap allowed: unclick the first option under overlap allowed:
 - o This will indicate that overlap is not allowed for the road concerning the material (a road is made out of one type of material), while the material can be used for more than one road.
- Relate the sentence with the classes:
 - o ... Allowed as 1st entity class: select “Road”
 - o ... Allowed as 2nd entity class: select “Material Type”.

Follow the same steps for Road type relation as follows:

- If you start from: 
 - o Click on “Relation Class”
- Click on “Add/Edit”
- Click on the empty line under “Relation Name” in the screen.
 - o Relation Name: Write: “Road is from type”
 - o Relation in sentence: Write “ is the road related with road type “
- Unclick the first option under “Overlap Allowed”:
 - o This indicates that one road can only have one category (road type), while a road type can be related to more than one road.
- Relate the sentence with the classes:
 - o ... Allowed as 1st Entity Class: Select “Road”
 - o ... Allowed as 2nd Entity Class: Select “Road Type”
- Go back to the Splash Screen / Home Menu , and refresh with <F5>.

Registering referencedata: Relate an instance to reference instances


- : Click on the subwindow “Entity Class” for “Road” and select “Entity Class Instances”
- Locate and select the A325 road, then click on “Details” to view its details
- Click on “Add Relation (select 2nd)”
- Click the drop down list and select “Road is from type” as the relation
- Choose “FW (Shortname)” to indicate that the A325 road belongs to the freeway road type.
- Close the window (Close (&Save)). Then open the tab “Reference Instances”
- As you see, the A325 road is related to the road type “Freeway”.
- Go back to the Splash Screen / Home Menu .

- Please try to register another road type for the A325, with an overlapping period. If an overlap is realized, the registration will not be accepted. This due to the setting in the relation tab for this relationrecord ().
 - o If you wish to register another road type, you must first specify an end date in the current road type record.
 - Click the “edit function” checkbox (at the bottom of the screen). This will make the edit button visible behind the relationship records. Click on the edit button for the appropriate relationship record and terminate the relationship (or delete the relationship) according to the instructions on the screen.
 - Afterward, you can enter the new road type. However, please note that the entry will only be accepted if there is no overlap in validity between the road types.

Now, please try to relate the road A325 to a material type, following the steps as elaborated above. Relate the road A325 to the material Asphalt.

Navigate

Now, let’s explore a bit by performing the following actions:

- : Click in subwindow Entity Class on “Place”, then click “Entity Class Instances”.
- Choose “Nijmegen”, click “Details”
- Open the “Relations” tab to observe that Nijmegen is connected to two roads.
- Open the road A325 within the same graphical user interface (GUI) to discover that it is connected to Nijmegen and Arnhem.
 - o To accomplish that, click on the magnifying glass icon next to the relationship record on the details screen.
- Open the place Arnhem, to see all details concerning the city of Arnhem.
- Via relations-tab; go back to the detailscreen of the A325 road.
- Through the reference instance tab, examine the type of road the A325 is, and the material from which this road is made
- Proceed to the Asphalt details and click on the “Relations” tab to access an overview of all roads that are constructed using Asphalt, which, at the moment, consists of only one road.

The way it works

This is essentially how it functions. You have the flexibility to define various aspects and delve into great depth. You can document the recipe for asphalt, the road's colour or speed limit, and record bridges and traffic signs. By using hectometre markers, you can divide the road into road sections. Furthermore, you can define the trees planted within a city or alongside the roads. Need to know the exact location? Record the longitude and latitude. When defining relationships, you can select multiple classes. For instance, when establishing the relationship "is the item having a tree," you can choose "Road" and "City" as the first entity class, and "Tree" as the second entity class. If you don't want to register each individual tree, you could consider selecting a "trees in city" class. You can associate the "trees-in-city" record with a specific type of tree, such as oak, beech, or chestnut, link it to a city, and record the quantity.

This approach allows us to gain insights into our corporate data model. We can register and establish relationships between various entities such as areas, colleagues, contracts, connections, production- and generation units, land parcels, market participants, market participant roles, payment condition types, and Boolean expressions (including MustRun, MustQualify, IsPerson). We can also create selection lists for payment conditions, status types, voltage codes, and more. Essentially, we can define any conceivable direct relationship between entities. Additionally, the tool's generic user interface enables seamless navigation between records, regardless of their content, thereby establishing insights in indirect relationships.

Our data storage method closely resembles Entity-Attribute-Value (EAV). We utilize proprietary numeric IDs that are not context-specific. This data storage approach lends itself well to analysis through graph technology/tripleStore.

As I see it, the next step will involve analyzing the data to extract its maximum value, once we have configured the TenneT wide data model.


Handiness for data editing and navigation

Navigation

Search via Id

Every instance is assigned a unique numeric ID, which is prominently displayed in the details screen. Additionally, you will find this ID in various other locations, such as the relations tab within the instance detail screen.

To directly access a specific instance from the start screen, you can use its ID.

: Simply enter the ID number in the "Instance Id" field and click on the "Go" button. This action will open the details screen for the corresponding instance, allowing you to view and modify its information.

Search on an attribute value

Throughout the application, there are multiple areas where you can perform searches based on attribute values. Simply enter the desired search term in the attribute value field.

If your search yields multiple results, an intermediate screen will open, displaying all the matching results. On the other hand, if your search generates exactly one result, the details screen for that specific result will open directly.

The search is not limited to specific attribute values. Any registered attribute value will be found during the search.

To further simplify the process, you can utilize wildcards (*) to broaden your search criteria.

Travelling through the data

Navigating through the data between the instances of different entities (classes) is the strongest aspect of this approach. If there is a relationship, whether direct or indirect, you can view all related instances through the different classes via a generic user interface. Navigation as if you are browsing the internet, clicking hyperlinks. Simply click on the "Relations" tab or the "Reference Instances" tab to access the related instances. And from such an instance, you can see how that instance is related to other instances from whatever class.

The editing of master and referencedata

The main focus of this article has been to explain the fundamentals of data modelling using the tool.

In terms of editing data, such as removing, modifying, or adding attributes and instances, I rarely use the graphical user interface (GUI). The GUI's functionality in this aspect is limited. However, for this demo model, I have incorporated functions that allow you to remove attributes, relationships, and instances. It's worth noting that there is room for improvement in these functions. They may currently be somewhat cumbersome to use. Nevertheless, they have been included to provide you with full control. Let's delve into the description of these functions below.

Some of the topics that will be covered have already been discussed in the previous section of the document

Add an attribute to an instance

- Open the detail-screen concerning an instance (like Place Nijmegen)
- Click Add Attributes
- All available attributes will be shown.
 - o An attribute is available if:
 - The attribute is related to the entity-class (via configuration as explained above) **and**:
 - The attribute has not yet been registered for this instance **or**
 - The attribute has been configured with “overlap allowed” = “Yes” **or**
 - The attribute has been registered for the entity, with a startdate and an end-date.
 - If overlap is not permitted: You can only register an attribute value for a period that does not result in overlap for that particular instance with respect to the same attribute. It is important to note that the system operates using validity periods, which include an “until” end date, rather than an “up to and including” end date.

Edit (or delete) an attribute-value concerning an instance

- Open the detail-screen concerning an instance
- Select an attribute
(tab Text Attributes / tab Quantity Attributes / Tab All Attributes / Tab Hyperlinks)
- Click “Edit”
- If the start date of the attribute value is in the past, you cannot simply modify the start date or the value. You can only change the end-date, if the end-date field is empty or the value is in the future. To make changes to the start date and/or the value, you need to enable “edit mode” by clicking on it. Once in edit mode, you can make any necessary modifications.
- Once you click the “edit mode on” button, a warning message will appear. Along with the warning, you will see a button labelled “delete” If you choose to click on the “delete” button, the entire attribute value record associated with that instance will be deleted. It is important to note that this action only removes the attribute value record and not the instance itself.

Delete a relation record concerning an instance

- Open the detail-screen concerning an instance
- Open the tab showing the relationship (Tab Relations or tab Reference Instances”)
- Click “Edit function” checkbox
- At every shown relationline, a new button appeared with the edit sign
- Click it, and confirm the deletion request

Delete an instance

- If you wish to delete an instance (the whole record), you must first delete all attribute-value records and relationship records associated with that particular instance. In other words, there should be no remaining connections or relationships linked to this instance. If any related data is not yet deleted, an error message will appear, indicating that the deletion cannot be completed until all related data is removed.

- To delete an instance, follow these steps:
 - o Delete all attribute value records associated with the instance, as explained previously.
 - o Delete all relationship records connected to the instance, as explained previously
 - o Once all attribute values and relationship records are deleted, the details screen for the instance will appear empty, showing only the ID of the instance.
 - o Click on the "Delete instance" button, to remove the instance.

Validation

Validate attribute values

It will be possible to validate input of attribute values. You can do this by entering an "MsAccess-like" formula. This functionality will be made available in a next version.

A formula that enforces the minimum and maximum number of characters of a name is:

```
IIf(IsNull("[VAR1]"),"",IIf(Len("[VAR1]")>4,IIf(Len("[VAR1]")<50,"","Use 30 characters or less (5 to 50 characters)","Use 5 or more characters (5 to 30 characters)"))
```

Relationships and overlap

When entering relationship types (🏠: Add/Edit Relation class), you have the options "overlap allowed 1st entity" and "overlap allowed 2nd entity." Depending on what you input, you may or may not, as the name suggests, connect instances with overlap.

This works as follows:

- Relation type: Car has colour
- Allowed as 1st entity: Car
- Allowed as 2nd entity: Colour

Car	Overlap allowed		What is allowed and what is not allowed
	1st	2nd	
1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<ul style="list-style-type: none"> - A car is allowed to have both the colour red and the colour green on the same moment. - Multiple cars are allowed to have the same colour at the same moment. - Any form of overlap is permitted.
2	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<ul style="list-style-type: none"> - A specific car can have the colours purple and yellow on the same day; overlap is allowed. - The car cannot have the colours red or green unless the colours red and green for car 1 have a start and end date that does not overlap with the assignment of one of those colours for car 2.
3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<ul style="list-style-type: none"> - A specific car can have the colour red on a given day. The car can have a different colour on another day but not at the same time. So, today it can be red, and tomorrow it can be a different colour. Overlap is not allowed. - Multiple cars can have the same colour. So concerning that; overlap is allowed.
4	<input type="checkbox"/>	<input type="checkbox"/>	<ul style="list-style-type: none"> - A specific car can have only one colour at any given moment. Another colour is allowed but not at the same time. Overlap is not permitted. - Each colour is assigned to a maximum of one car. You cannot have two green cars at the same time. Any form of overlap is not allowed.

Attributes and overlap

In the attribute configuration screen, you can specify per attribute whether overlap is allowed. If overlap is allowed, you can register an attribute value for instances of the related classes with a time overlap. If overlap is not allowed, you can only register a new attribute value for that attribute if this new input does not cause any overlap with other attribute values of the same attribute over the same instance.

Storing and accessing information outside of data registration

Metadata

From a governance and organizational perspective, it is desirable to capture additional metadata alongside all entities, attributes, relationship types, and their interrelationships. However, I consider the registration of meta data (which is crucial for a proper governance) secondary to the actual management of master data. Master (and reference) data registration is the primary function facilitated in this environment.

To ensure that necessary meta-information (the part that is not automatically generated by configuration) can be registered, stored and accessed in a way that doesn't overwhelm users within the tool unless they explicitly wish to look it up, there is the option to reference internet pages via hyperlinks. This allows for efficient access to supplementary information when needed.

In the application, when registering classes, you have the option to add a hyperlink address. By adding a hyperlink, you can directly access the page associated with this hyperlink address. Both in the list view of instances (select a class, click Entity Class Instances) and in the details screen of an instance, you will see Class names underlined. If you click on a class (and a hyperlink address is registered for the respective class / attribute), the corresponding page will open.

Make sure you enter the hyperlink address in the following format:

```
http://www.<address>.<com/nl/gov/eu/etc.>/specification
```

On these pages, you can record all relevant metadata, but also meeting notes regarding decisions made about the class/attribute. While it may not have immediate functional value, when you need to integrate complex data models into a single enterprise data model, it becomes invaluable to have information such as decision-making processes and thought processes documented.

External resources

You can link an instance to one or multiple web pages. For instance, if you have a dossier in an external environment for each instance, and this dossier can be accessed via a hyperlink address, you can enter this. You register an attribute, for example, "Dossier" with the data type Hyperlink. Then, you associate this attribute with the class(es). Next, when registering an instance of a class, you enter the file by inputting the hyperlink address. Subsequently, you can view all internet addresses related to the instance through the details view, in the "Hyperlinks" tab, and open them using the "open link" button. Make sure you enter:

```
http://www.<address>.<com/nl/gov/eu/etc.>/specification.
```

The address <http://www.tennet.eu/about-tennet> will be accepted.

Bulk upload & working directly in the database

The primary purpose of this application is to facilitate the reverse engineering of data models. When it comes to inputting bulk data, I prefer to work directly in the database. Therefore, it is essential to have a thorough understanding of the database knowledge. Appendix 3 explains the tables and relationships between the tables of this EAV concept.