Now, entering field name in the table.

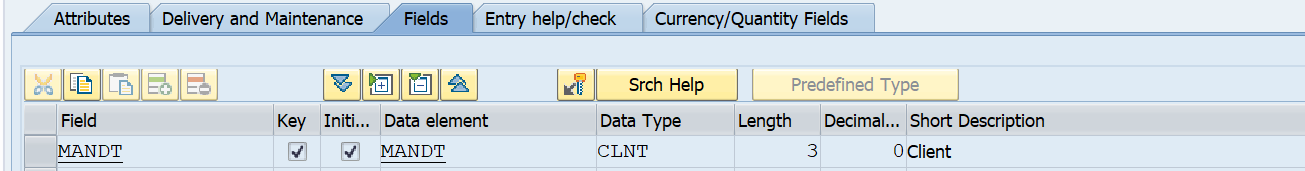
Firstly, our field names can start with any letter at all. They do not have to begin with a Z or a Y like our table names themselves. The field names can have a maximum of 16 characters.

And our tables must have at least one key field. They identify each record as being a unique record.

You can assign an initial value to a field.

Every field that we use in our table is made up of what we call a data element. And the data element defines specific attributes of each field. One word of warning here, data elements must adhere to the customer namespace again.

So, they must, for anything new that we create, begin with the letter Z or Y.

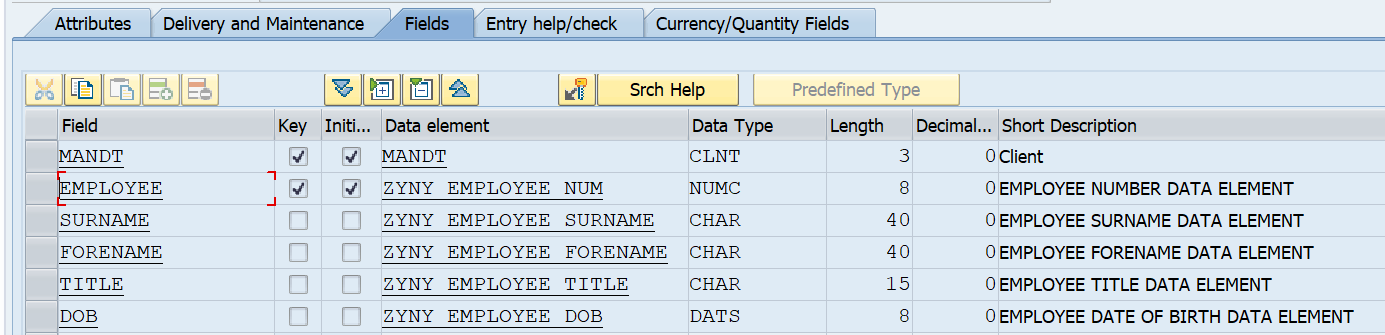


So, let's start off with entering our first field, and the field we're going to create is a very important field within an SAP system. It’s the field that identifies the client that our records are associated with. So, in the field name enter client. In the data element we will enter MANDT. Now the data element already exists in the system. And when I press enter after entering it you can see the system automatically filled in the data type, the length, number of decimals, short text for the actual data element itself.

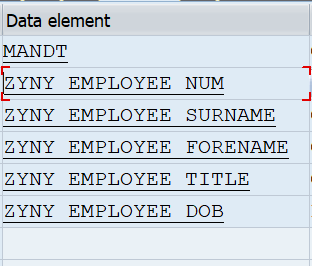
Now, very important, we need to ensure that the client field is made a key field within our table.

The next field we're going to enter is going to be our own created field. And this is going to be employee. Again, we're going to make this field a key field.

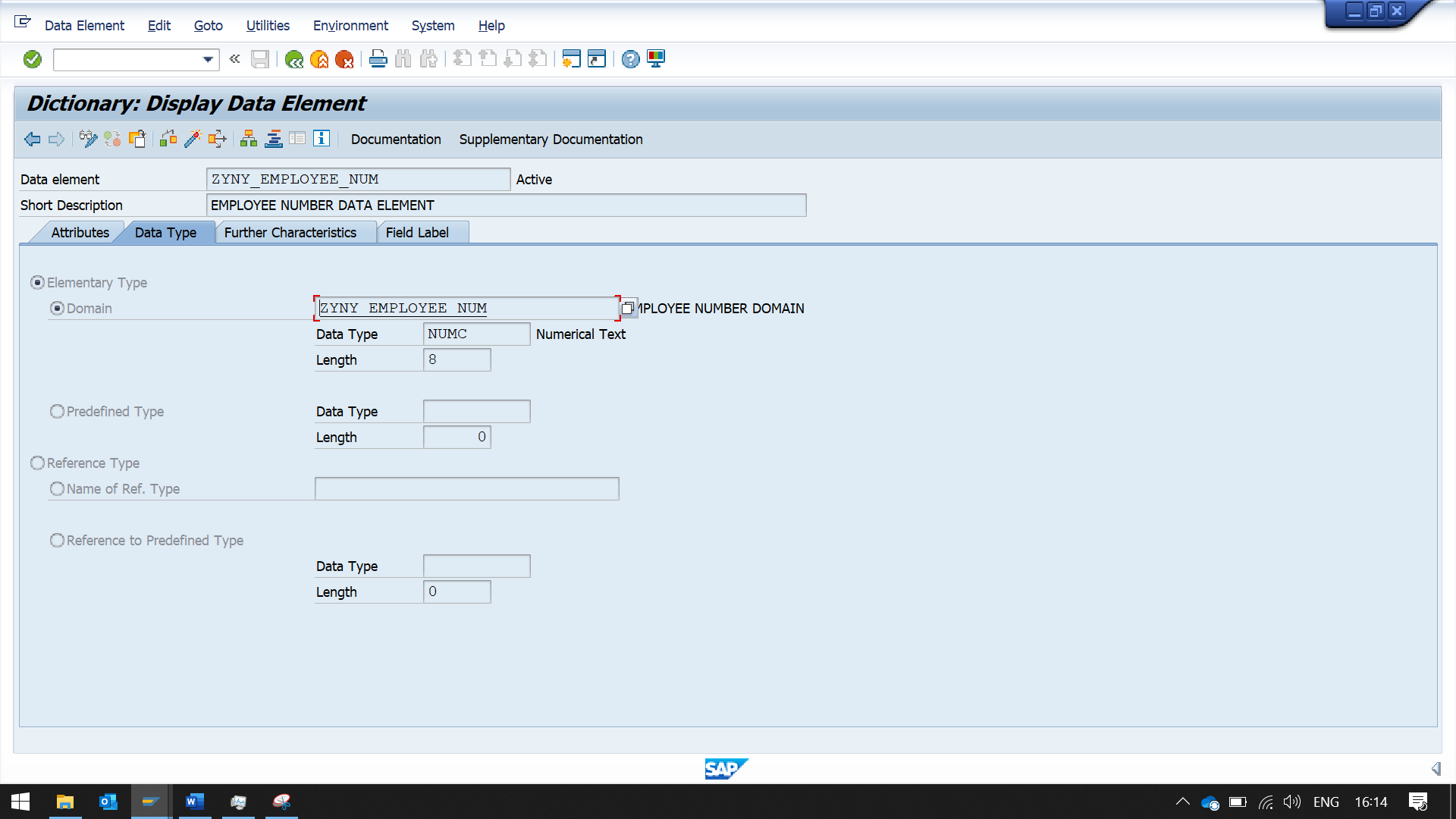
click the Save button.



we're going to use forward navigation to create the data element. Another way of doing it is to open a brand-new session. Go to SE11 again and start creating the individual objects this way.



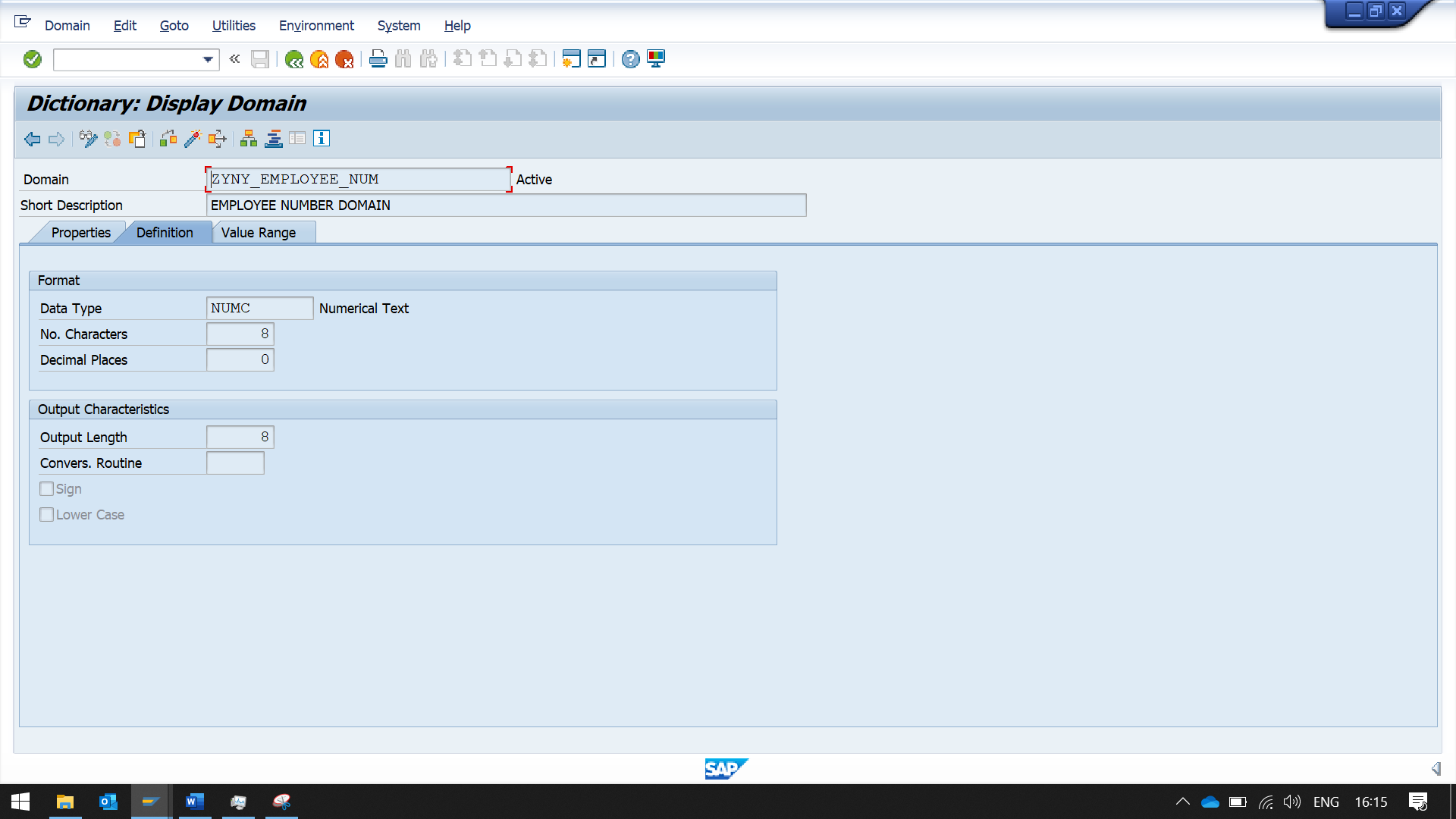
we want to use forward navigation. And we do this just by double clicking our brand-new data element. So, we get this window that appears called create data elements and it’s telling us the element does not exist do we want to create it, answer yes.



Then we get taken to the Maintain Element Window. The Data Element window is asking us for some short text, so let's enter in here, Employee Data Element. Now under the short text you can see four tabs. And the one that's highlighted says Datatype. We can see here the Elementary Data Type, called the domain, needs to be defined for our data element.

Domains must adhere to the customer namespace. Now we're going to create a brand-new domain using the same name we use for the data element.

And once again we're going to use forward navigation to create the domain. So double click the entry. It’s telling us our data element has changed and do we want to save it before creating our domain. Choose yes.



We get the create object directory entry window appearing again. And just as before, we want to save this development to the door temp development class. So, you can just click the save button. So, now because we've used forward navigation, it's telling us that the new domain we are trying to create does not exist. Choose yes to create the domain. And once again, enter some short text that describes what the domain is going to be used for. So, we will just enter ZYNY\_EMPLOYEE\_NUM domain. Under the short text, we can see three tabs and the one that's highlighted says definitions.

The fields available to us, the first one says data type. Click the drop down on data type. You will see lots of generic data types already exist within the ABAP dictionary and we can use one of these for our employee field or domain. We’re going to choose NUMC, which is a character string with only 8 digits. So double click the entry and that fills in the data type field for us. In the number of characters, enter eight which means this field is going to contain a maximum of eight characters. And we want zeros for decimals, which is the default anyway for a NUMC field. Select an output length of eight. And just hit the enter key.

We can see here that where we have entered the NUMC, it has brought back the description of this field, which tells us it's a valid entry. so, we all need to do is click the save button.

Once again, we are presented with the object data entry pop-up box and we want to assign this object as a local object, so click the local object button. Now the next thing we want to do is activate this object. Even though we have saved it, it doesn't mean we can actively use it within our table yet. We need to set the object as active, which means our other data elements can use this domain going forward.

So, on the toolbar, you'll see a little matchstick icon. You hover over, it says activate. And you can also press the Control and F3 key. Click the activate button so what we need to do is click the green tick. The system goes away, checks all the entries we have made for the domain itself, and if everything is okay, it will then activate the object. And if you look at the status bar at the bottom of the screen, everything's okay, so we can now proceed on with creating our table.

Now remember, we used forward navigation for generating our domain. So, what we need to do is step back. So, click the F3 key, and you are taken back to your data element maintenance screen. Now because we have created and activated our domain, we can now see the text for our domain has come through and been placed, assigned the ZEENUM entry that we entered before.

And this has brought back individual domain properties that we created. Now the next thing we need to do is define the field labels. So, click on the field labels tab and here we need to define the short, medium, long and heading field labels for our data element.

Now you'll see on the left-hand side there’s a length field which I have on purpose not filled in. Because once you put the entries in the field label, and press the Enter key, it will automatically calculate the length of the entries that you have typed in.

Once this is complete, save the data elements and then activate it. Again, we get a window showing the inactive objects.

At last, make sure domain, data element and table all are activated. 



