Master-Detail Table to NRDM

Summary
In this example there is a SALESORDER table and its SALESITEMs. The endresult should be an XML structure with
+-SALESORDERs and their attributes
+-SALESITEMs of this order.
So we want a nested structure where each SALESORDER contains many SALESITEMs as children.

At a first glance, the corresponding DataFlow looks like a join of the two source tables.

SALESORDER(source_ds.SOURCE)

| | 
|-----------------|-----------------|
| Query_NRDM_Join | 

SALESITEM(source_ds.SOURCE)

| | 
|-----------------|-----------------|
| But as we said NRDM means dealing with table datatypes. So at the root level, how many rows do we want to see in the output? Exactly as many as SALESORDERs. Do we need any SALESITEM information at the root level, no, these are all child attributes. So, we just said we want to select from SALESORDERs only at root level - we have to configure it then. We go to the from clause of the query and remove the SALESITEM table that was added by default. |
Now we can build the root level in the output by drag and drop all columns from SALESORDER we need. Please pay attention, you cannot drag columns from SALESITEM as we do not select from them. You would not expect to map columns from tables you have not selected in the relational model, same here.

In addition we add one more column of type table - we call it a “schema”. You give that column a name like you do with any other, only the mapping will look different.
As this is a "table" column, we again need to specify the select. But for this element. So view the "From" tab of the query and then right-click the schema column and say "Make Current". With that, this level now becomes active and we can specify the entire select. The "From" is empty by default. What kind of data do we want to see at that level? All SALESITEM information. So drag the SALESITEM table into the "From" clause. And how many rows do we want to see for each SALESORDERS? All SALESITEMs that belong to the current SALESORDER only, not the entire SALESITEM table for each order. In other words, we need a where clause as well.
As the "from" clause is now the SALESITEM, we can take all columns to this schema. We do not need the SALES_ORDER_NUMBER column in the child as well, as the relationship was defined in the where clause already.

SALESITEM.SALES_ORDER_NUMBER = SALESORDER.SALES_ORDER_NUMBER
To see the result, the easiest way is to add a Template XML - a XML file generated based on its input - just like a Template Table.