

Relational to XML Document Example

Objective

This example demonstrates how to map relational data to mapping classes in an XML document model which is based on an XML Schema (XSD).

Setup and Assumptions

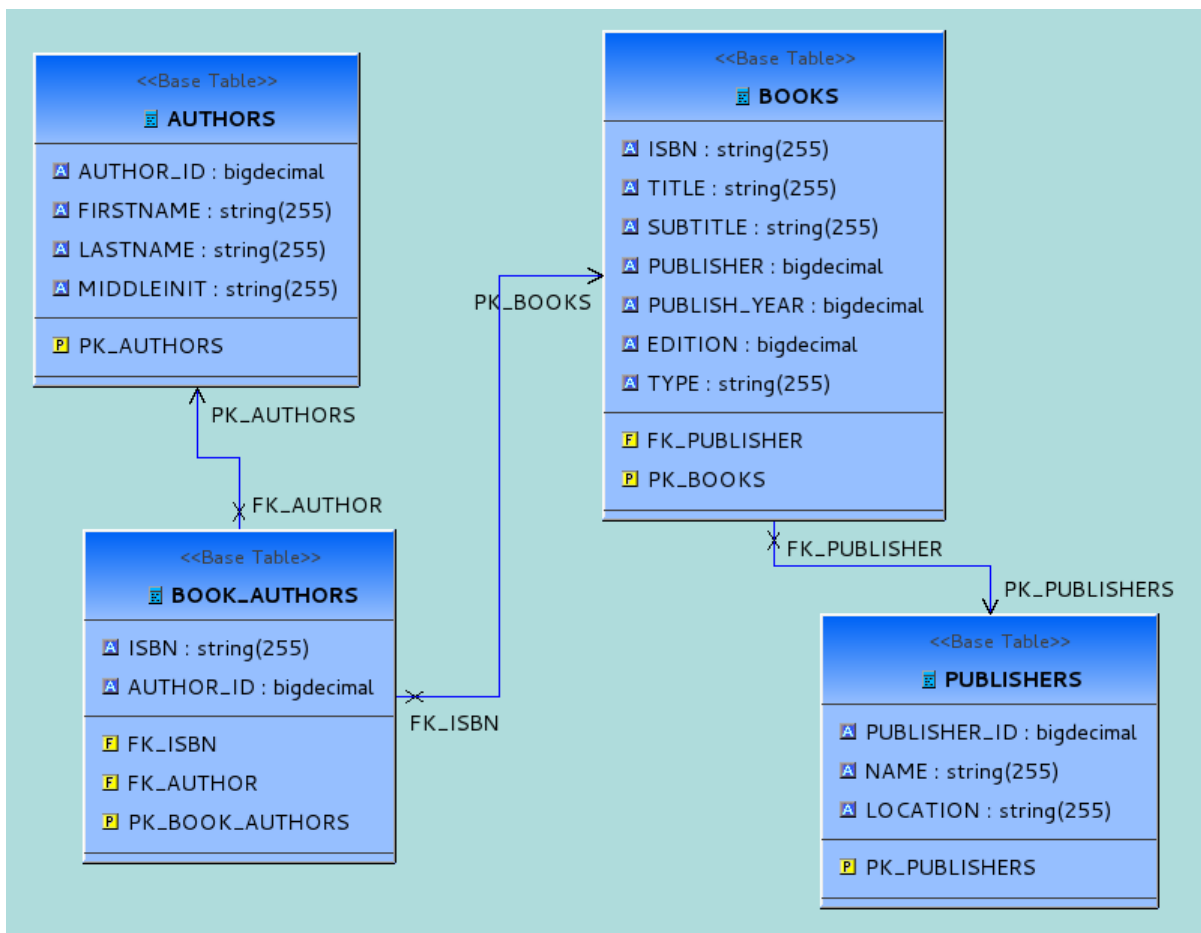
- The Teiid Server and Teiid Designer are installed and are compatible versions
- The Teiid Server is running and you have both admin and user access
- You have an XML Schema Document (XSD) from which you will build your own document model
- You have access to the relational source(s) used to populate the XML document
- Otherwise, this example assumes you are starting with essentially nothing

1. Start the Teiid Designer

- 1.1. Establish a connection to the Teiid server to which you will deploy the VDB
- 1.2. Add a Teiid Model Project and give it a name of your choosing

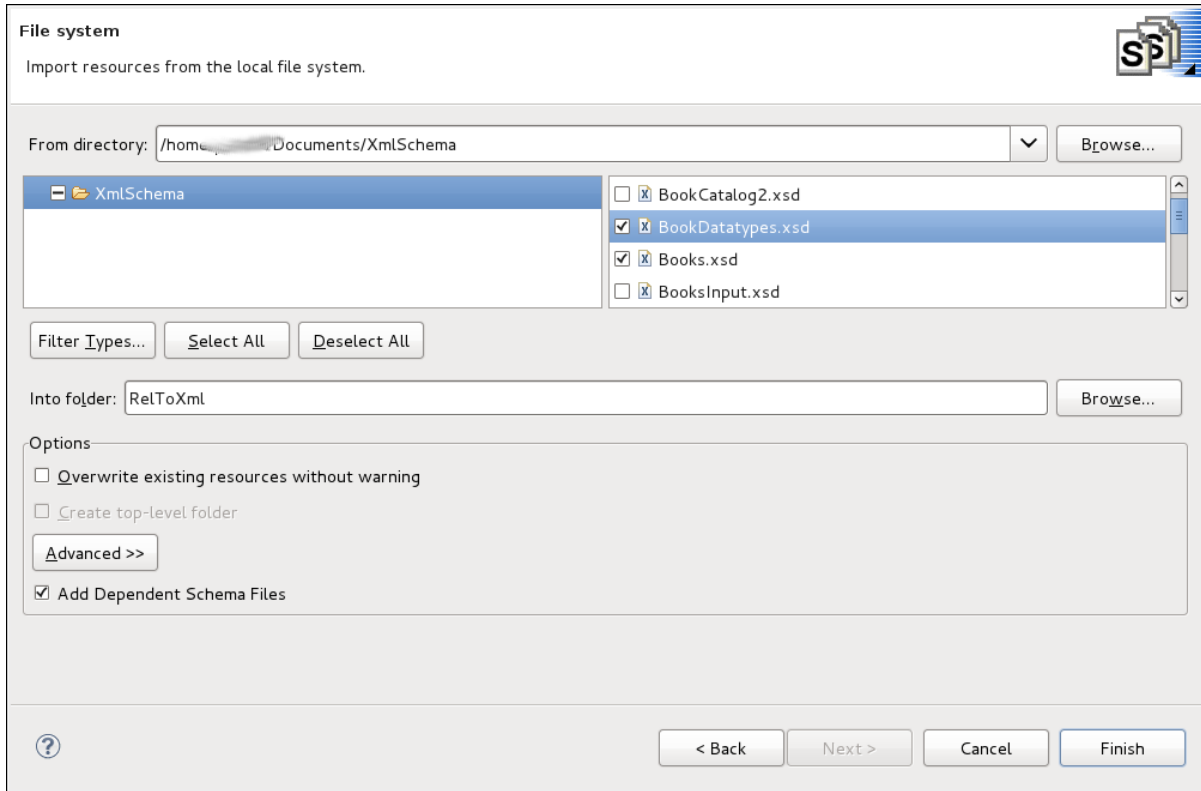
2. Import Metadata from JDBC

This example requires a relational datasource. Our example uses one named “Books”.

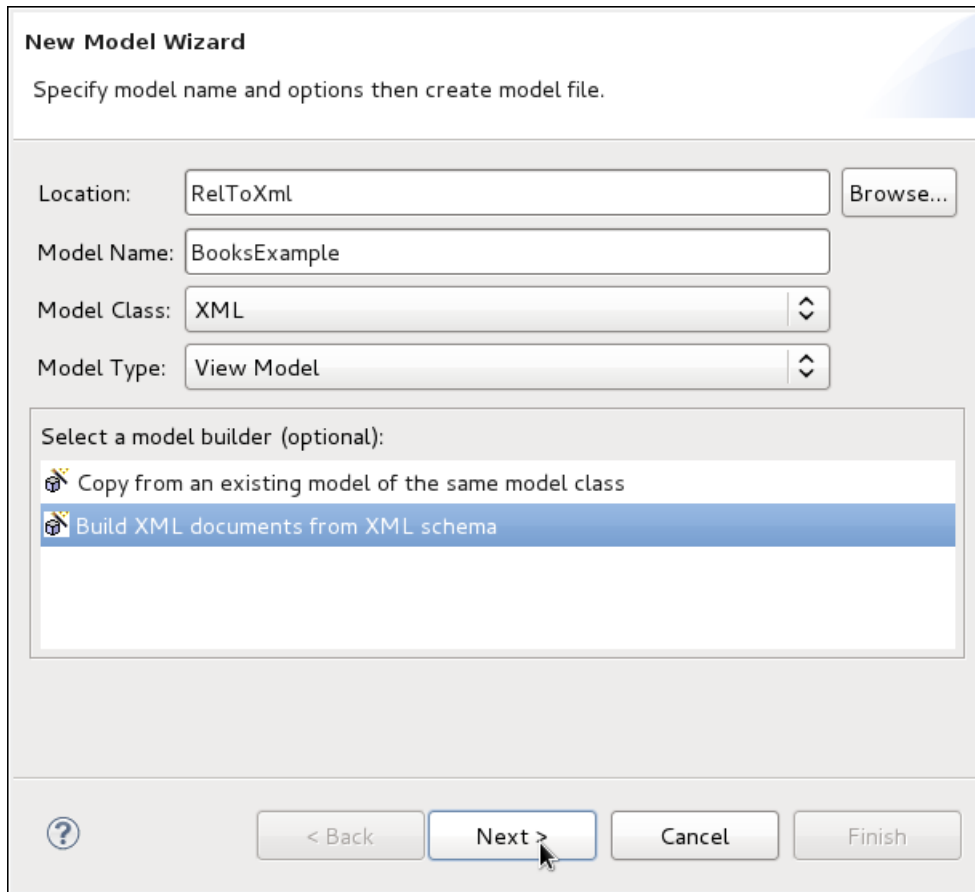


3. Import the XML Schema Document

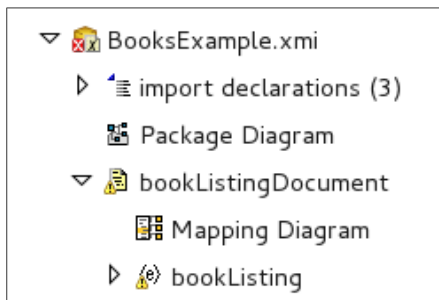
Our example imports one from the file system



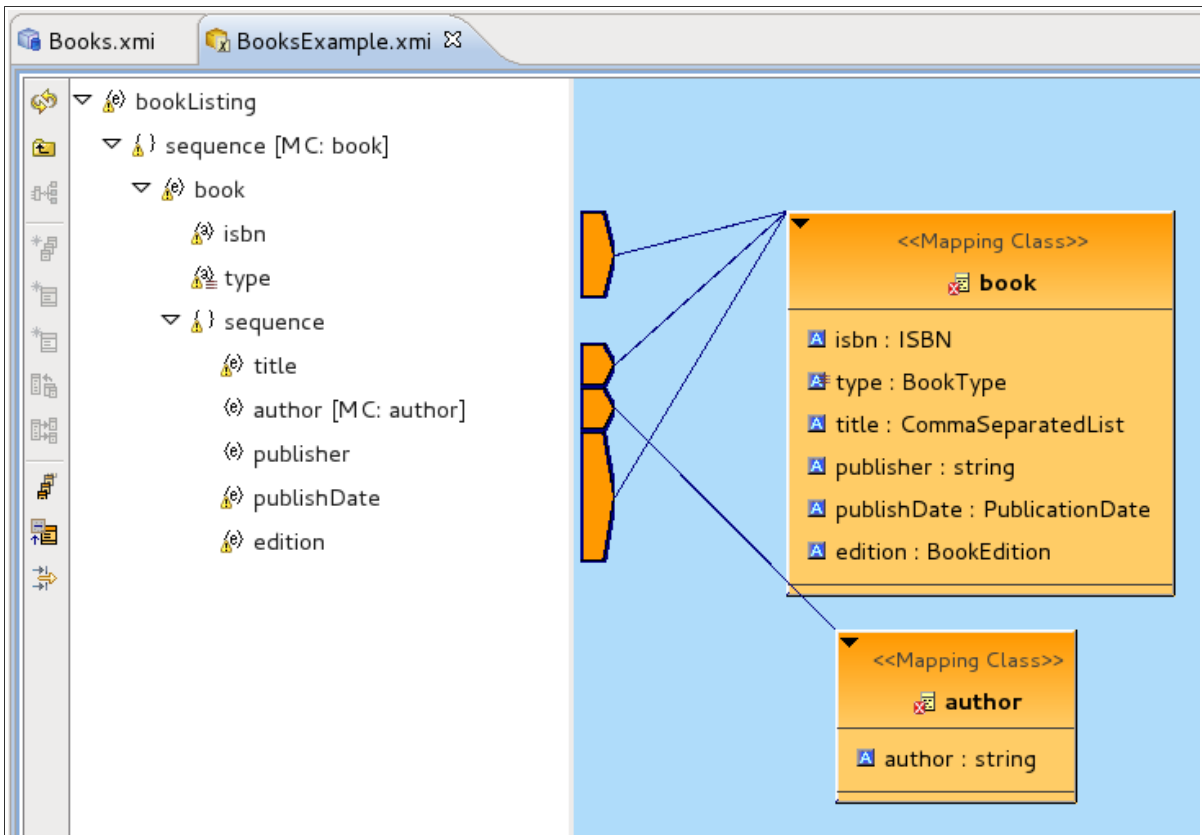
- ### 4. Create the XML document model
- 4.1. File > New > Teiid Metadata Model
 - 4.2. Give the model a name
 - 4.3. Specify the XML Document Model Class
 - 4.4. Select the optional builder to use your imported XML Schema as the basis for the document model



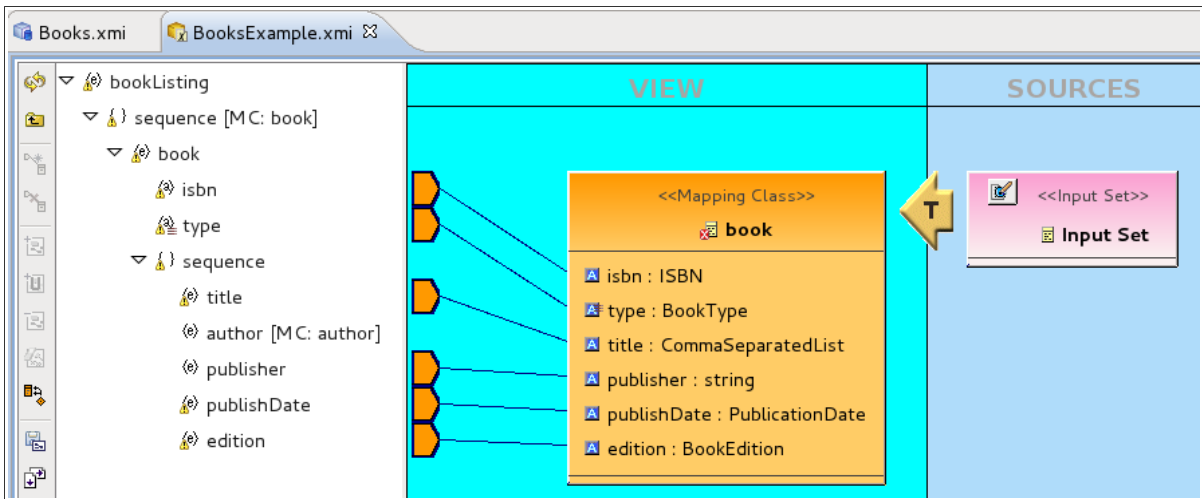
- 4.5. Click Next
 - 4.6. Select the XML Schema you imported
 - 4.7. From the list of Schema Root elements, select the one(s) you want and move them to the list of Virtual Documents
 - 4.8. Choose any other options you may wish to use
 - 4.9. Click Finish to create the document model
5. Mapping Relational Data to the XML Document Model
 - 5.1. Expand your model in the Model Explorer Tree



- 5.2. Double-click the Mapping Diagram object in the tree to open the Mapping Diagram
This mapping diagram shows the overall document model.

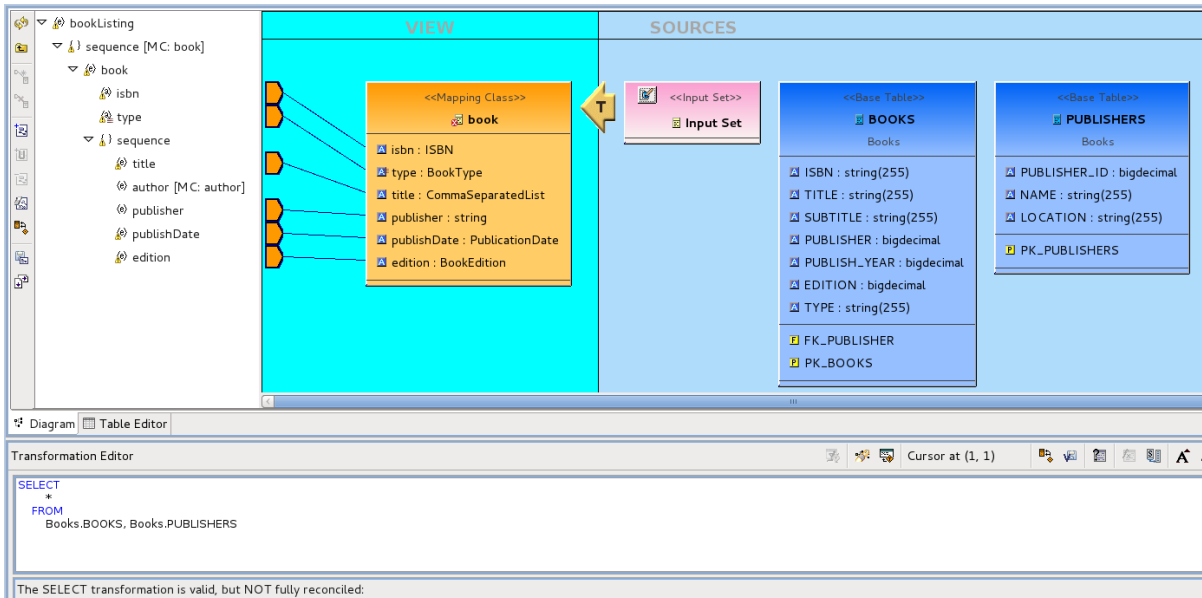


5.3. Double-clicking the book object opens its transformation diagram

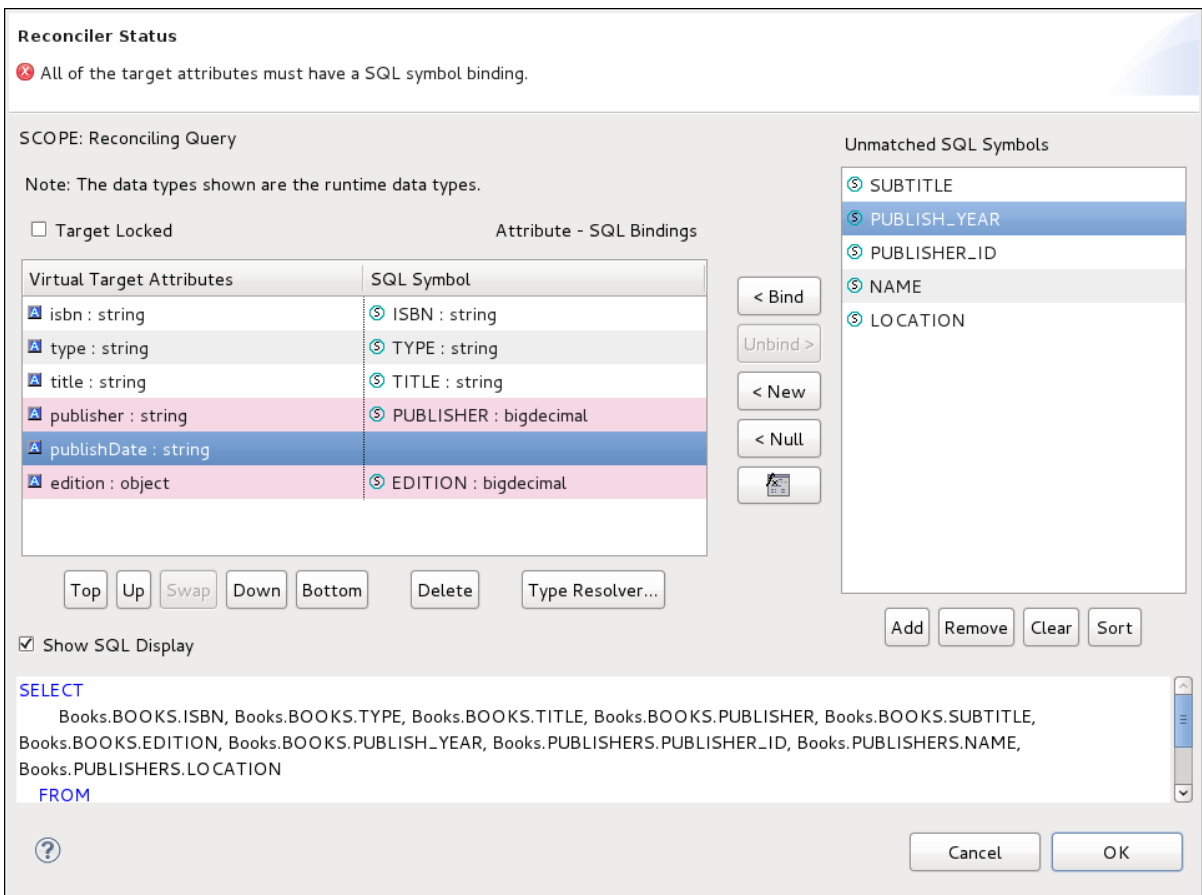


5.4. Add Tables to the Transformation

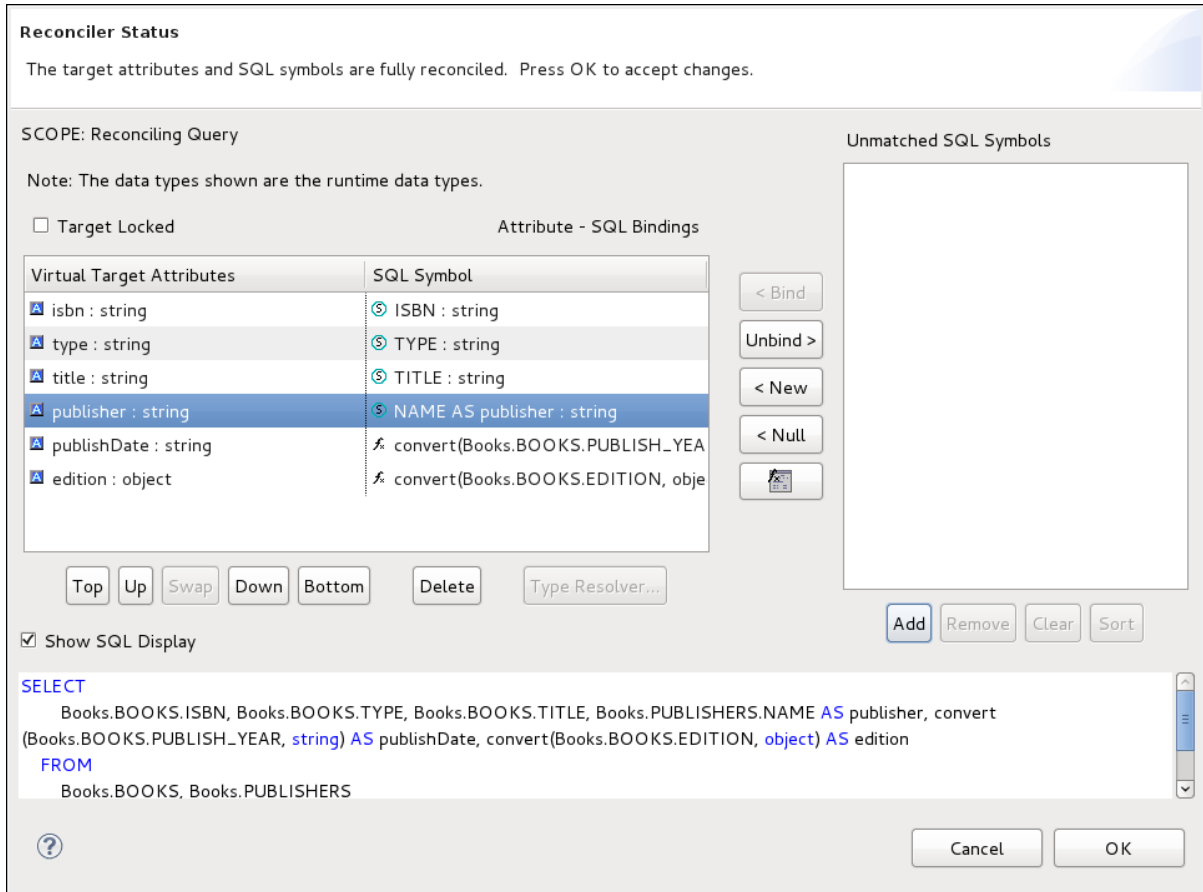
Just as you would when modeling a purely relational view model's transformation, add the source tables you need



- 5.5. Add any WHERE criteria which may be required to the transformation SQL
- 5.6. Save the model
- 5.7. Start the reconciler so the relational columns can be mapped to the mapping class attributes



5.8. Map the data according to the requirements of your models and data, resolving any data type conflicts and removing unused SQL symbols

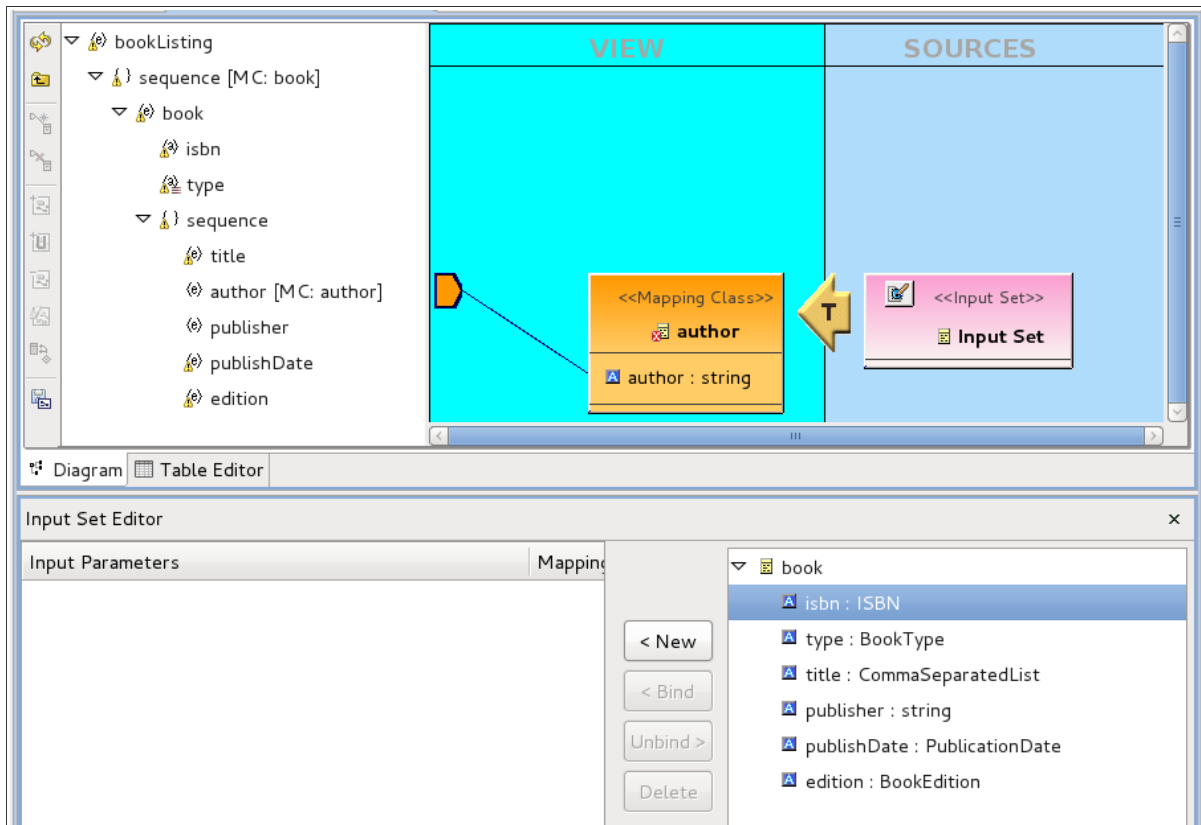


5.9. Return to the overall mapping class diagram, and open the transformation diagram for the next mapping class. In this example, that's Author

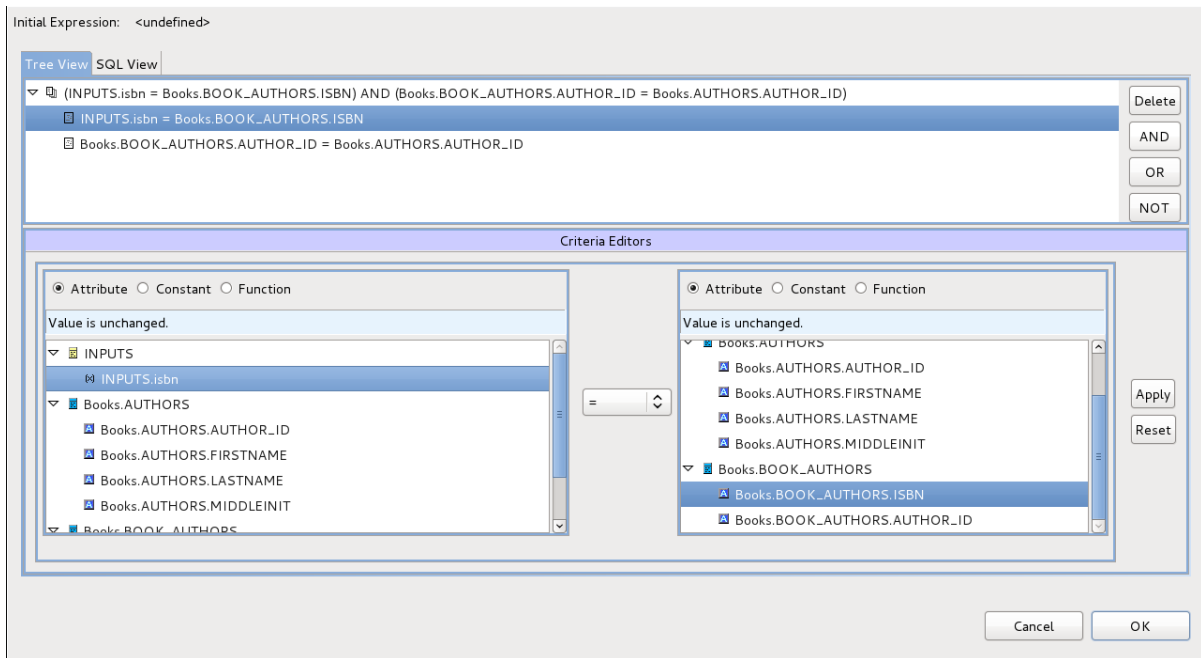
5.10. To tie this mapping class with the Book mapping class involves the input set. Open the input set editor by clicking the input set button



5.11. Select the attribute from the Book class that will be used to link the two mapping classes (in this case so the right authors are shown for each book)



- 5.12. Click the New button
- 5.13. Double-click the T node in the diagram to close the input set editor and open the transformation editor for the author mapping class
- 5.14. Add the relational table necessary for the transformation
- 5.15. Open the criteria builder
- 5.16. The input set parameter is available in the builder. Use this, and the other tables, to build your criteria



- 5.17. Click OK to exit the criteria builder and return to the transformation
 - 5.18. Save the document model
 - 5.19. Use the reconciler to make any adjustments to the transformation, and to map columns to the attributes
 - 5.20. Save all the models
6. Create a Virtual Database (VDB)
 - 6.1. Right-click the document model in the Model Explorer tree > New > Teiid VDB
 - 6.2. Name the VDB
 - 6.3. Click Finish
 - 6.4. Deploy the VDB
 7. Querying the Document Model

Once deployed, you can query the document model using any JDBC tool. Here's a simple query which returns the entire document's contents

```
SELECT * FROM BooksExample.bookListingDocument
```