

Qdabra DBXL

How to Create Database Mapping

 01/07/2018

Overview

Purpose	1
Pre-requisites	1
Create a SQL Table	1
Configure Database Mapping in DAT	4
Add Database Mapping for Nested/Repeating Group	11
Add Mapping for the Child Table.....	12
Conditional Shredding	15
Mapping DBXL Tokens	16

Purpose

Qdabra Database Accelerator, or DBXL, allows users to create mappings from your XML forms to your SQL tables. This document will guide you through the basic database mapping.

Pre-requisites

This document assumes the following:

- Qdabra DBXL (version 3.1 or later) is installed
- Microsoft SQL Server Management Studio is installed
- SQL Database is created
- DBXL DocType is created
- InfoPath form that submits data to your DBXL DocType.

For this tutorial, we'll make use of Expense Report form as a sample.

Create a SQL Table

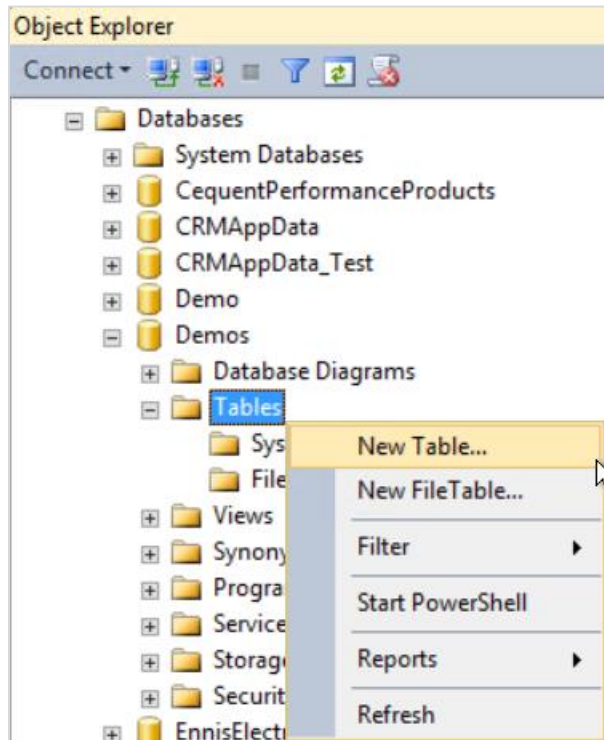
First, you will need to add a table in your already created SQL database.

1. Open SQL Server Management Studio and connect to the SQL instance.

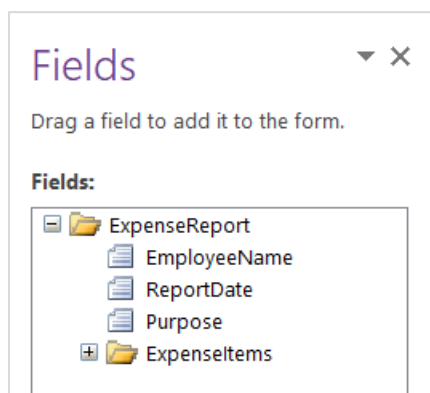
Note:

It is recommended to use the same SQL instance that was used to install the QdabraDBXL and QdabraUtility but we don't recommend adding tables to these databases that DBXL uses.

- Click the database you already created. In this guide, we have **Demos** as our sample database. Right click the database and select **New Table**:



- Recall your InfoPath form template's schema. Here we have **Employee Name**, **Report Date**, and **Purpose** fields. You may have more or less than three fields in your form.



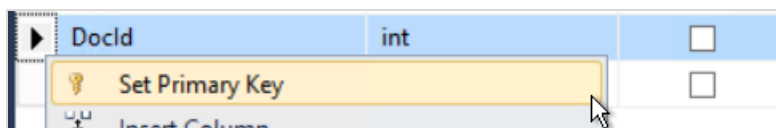
4. Create the corresponding columns for these fields in the newly created table.

Column Name	Data Type	Allow Nulls
EmployeeName	nvarchar(50)	☒
ReportDate	date	☒
Purpose	nvarchar(50)	☒

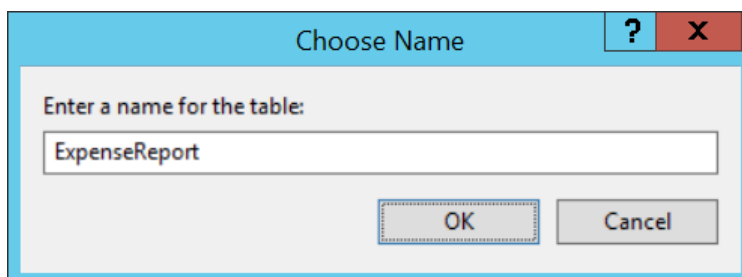
Note:

Your table can have a few or many columns. It depends on which data from the form you wish to store in SQL.

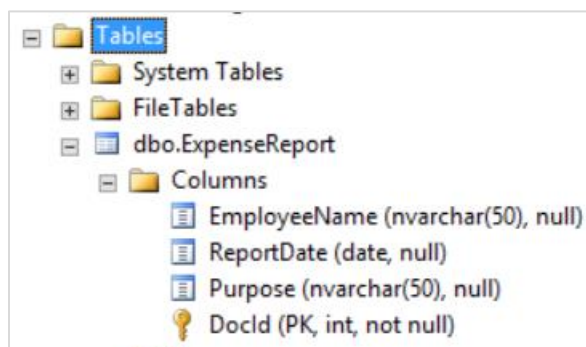
5. Add an additional column called **DocID**, with data type **int**, and uncheck **Allow Nulls**. Set this column as the **Primary Key**.



6. Save the table with any name. In this example, we'll call it **ExpenseReport**.



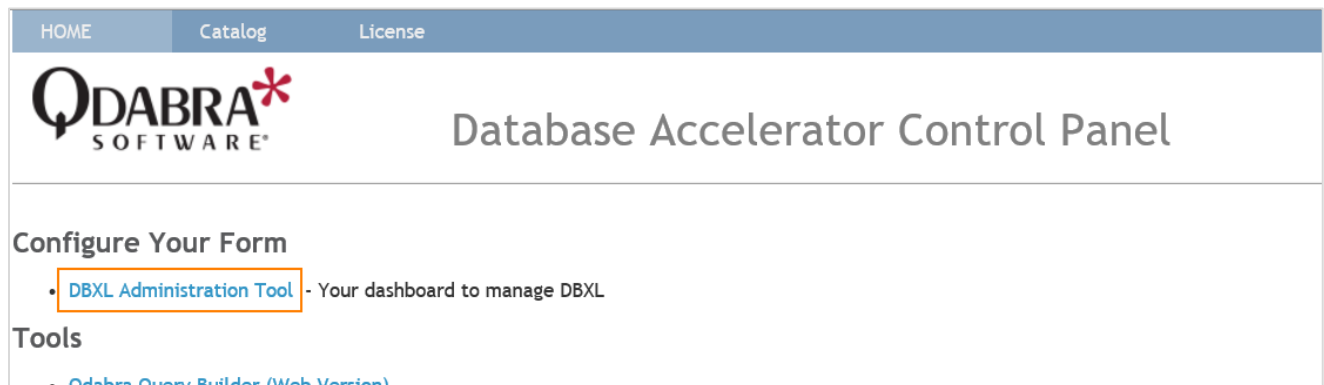
Here are the final table columns:



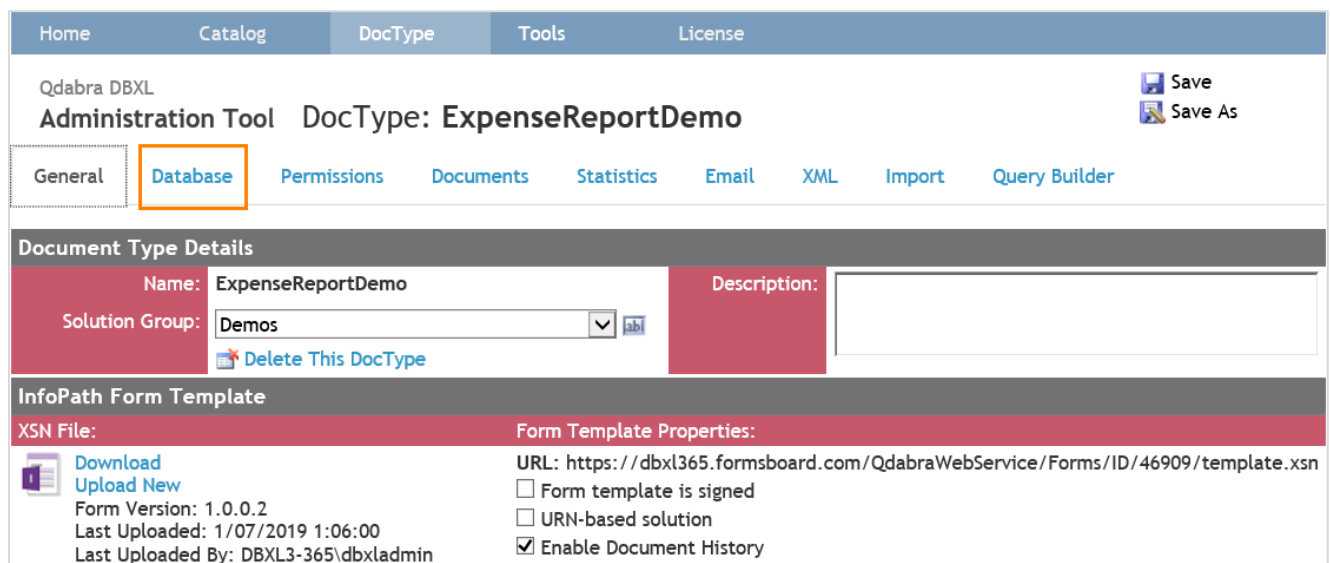
Configure Database Mapping in DAT

The following steps will guide you on how to setup the database mapping via the Database Accelerator Administration Tool (DAT) for the published form template. Our goal is to collect submitted information from a non-repeating group in the form into a corresponding row in the table.

1. Open the **DBXL Administration Tool (DAT)**.



2. Select the DocType that you want to edit.
3. Click **Database** in the DocType Menu.



4. In the **Connection String** field, enter the string appropriate for your SQL server. Example:
 Data Source=(LOCAL)\SQLEXPRESS;Initial Catalog=Demos;Integrated Security=True

Home
Catalog
DocType
Tools
License

Qdabra DBXL
Administration Tool
DocType: ExpenseReportDemo
Save
Save As

General
Database
Permissions
Documents
Statistics
Email
XML
Import
Query Builder

Database Configuration

Connection String:

Data Source=(LOCAL)\SQLEXPRESS;Initial Catalog=Demos;Integrated Security=True
Test Connection

(for example, Data Source=(LOCAL);Initial Catalog=TestDB;Integrated Security=True)

Submit Options:
Database Actions:

☒ Report Shredding Errors to the Form User
Reshred All Documents
Refresh Database

Database Mapping

SQL Side
Template Sides
Del

Table Name(Columns)
Column Name
Type
Key
Node Path

Add Table

powered by Qdabra DBXL
Copyright © 2016 Qdabra Software. All rights reserved. Qdabra and the Qdabra logo are trademarks of Autonomy Systems LLC. www.qdabra.com
English

Tip:

If you don't know the connection string, you can modify any one of the existing document types (installed by DBXL) and copy the connection string. You can then update this string for your purposes.

- Click **Test Connection**. Once successful, you'll see the following popup window.

Home
Catalog
DocType
Tools
License

Qdabra DBXL
Administration Tool
DocType: ExpenseReportDemo
Save
Save As

General
Database
Permissions
Documents
Statistics
Email
XML
Import
Query Builder

Database Configuration

Connection String:

Data Source=(LOCAL)\SQLEXPRESS;Initial Catalog=Demos;Integrated Security=True
Test Connection

(for example, Data Source=(LOCAL);Initial Catalog=TestDB;Integrated Security=True)

Submit Options:
Database Actions:

☒ Report Shredding Errors to the Form User
Reshred All Documents
Refresh Database

Database Mapping

SQL Side
Template Sides
Del

Table Name(Columns)
Column Name
Type
Key
Node Path

Add Table

powered by Qdabra DBXL
Copyright © 2016 Qdabra Software. All rights reserved. Qdabra and the Qdabra logo are trademarks of Autonomy Systems LLC. www.qdabra.com
English

Database

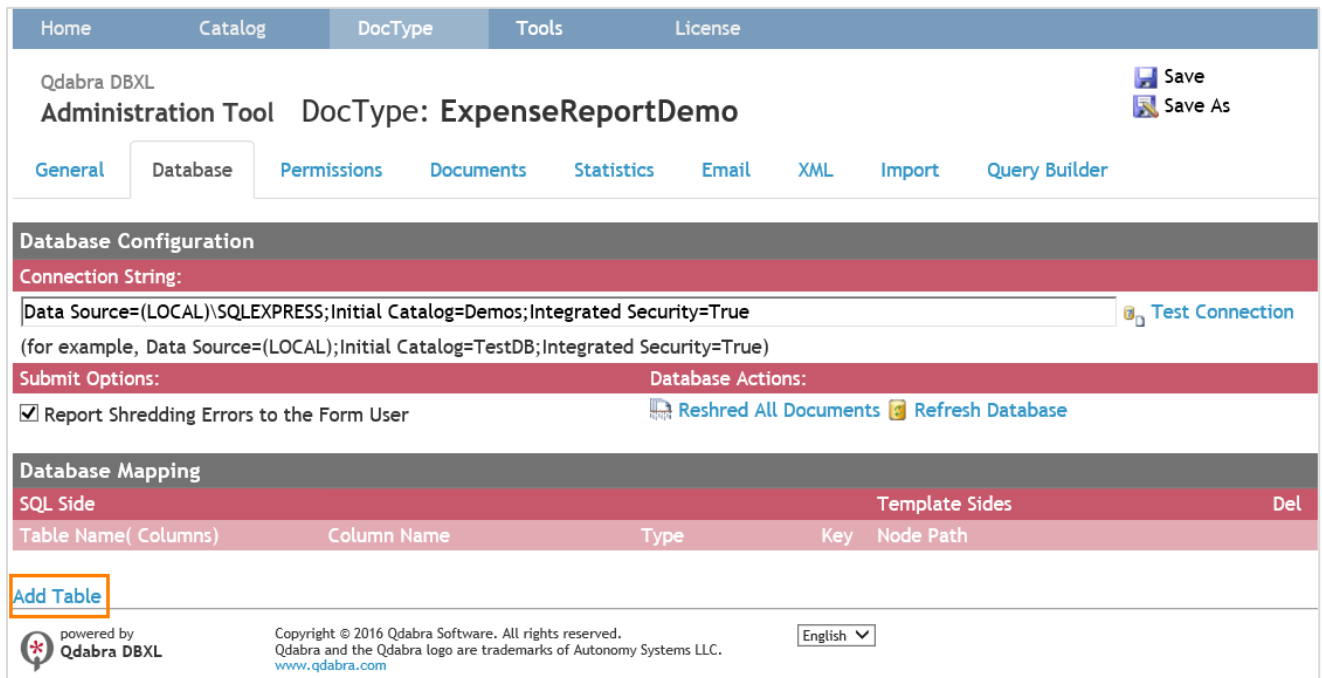
Connection was successfully established.

OK

Note:

Should you get a permissions error, the easiest way to troubleshoot the error is to compare the Security of the QdabraDBXL and your own database. Your database might be missing permissions. Simply refer to the QdabraDBXL database as a reference of users and permissions to add.

6. Click **Save** and then click **OK** in the confirmation dialog. Now you can construct the database mapping.
7. Click **Add Table**.



Home Catalog DocType Tools License

Qdabra DBXL Administration Tool DocType: ExpenseReportDemo

Save Save As

General Database Permissions Documents Statistics Email XML Import Query Builder

Database Configuration

Connection String:

Data Source=(LOCAL)\SQLEXPRESS;Initial Catalog=Demos;Integrated Security=True Test Connection

(for example, Data Source=(LOCAL);Initial Catalog=TestDB;Integrated Security=True)

Submit Options: Database Actions:

☒ Report Shredding Errors to the Form User Reshed All Documents Refresh Database

Database Mapping

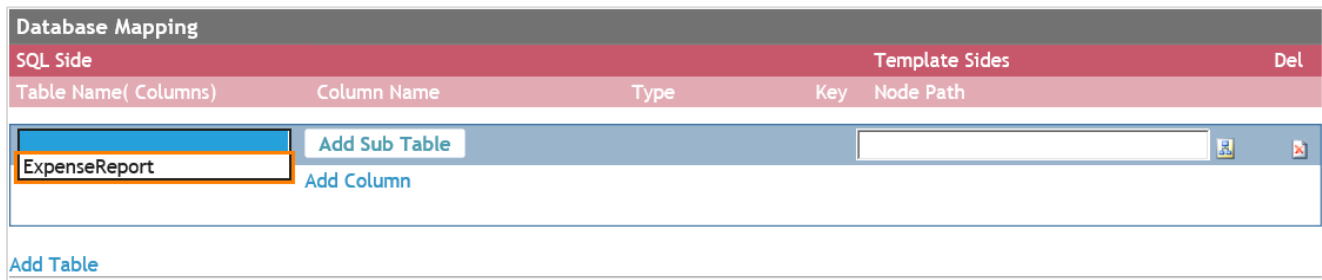
SQL Side	Template Sides	Del
Table Name(Columns)	Column Name	Type
Key	Node Path	
Add Table		

powered by Qdabra DBXL

Copyright © 2016 Qdabra Software. All rights reserved. Qdabra and the Qdabra logo are trademarks of Autonomy Systems LLC. www.qdabra.com

English

8. Select the correct table in the dropdown.



Database Mapping

SQL Side	Template Sides	Del
Table Name(Columns)	Column Name	Type
Key	Node Path	
ExpenseReport		

Add Sub Table

Add Column

Add Table

9. Click **Add Column** and select one from the dropdown.

Database Mapping				
SQL Side			Template Sides	
Table Name(Columns)	Column Name	Type	Key	Node Path
ExpenseReport				
Add Sub Table Add Column				

Database Mapping				
SQL Side			Template Sides	
Table Name(Columns)	Column Name	Type	Key	Node Path
ExpenseReport				
Add Sub Table DocId EmployeeName Purpose ReportDate Select...				

10. Repeat the same for the rest of the columns.

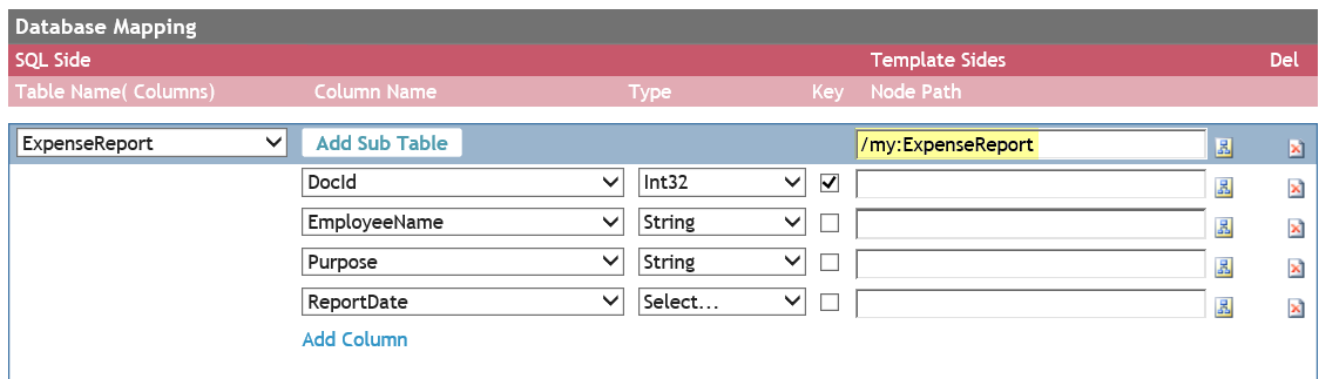
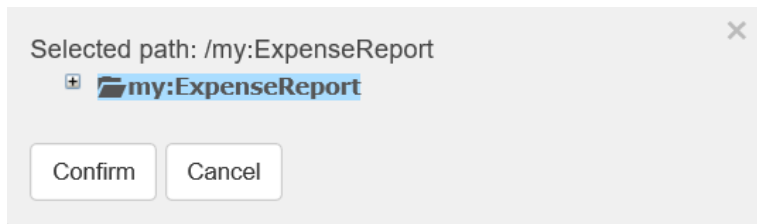
Database Mapping				
SQL Side			Template Sides	
Table Name(Columns)	Column Name	Type	Key	Node Path
ExpenseReport				
Add Sub Table DocId EmployeeName Purpose ReportDate Int32 String String Select... ☒ ☐ ☐ ☐				

The steps above added the “destination”, that is, the SQL table and columns where the data will be stored. Now you will add the “source”, that is, the InfoPath form template nodes where the data comes from.

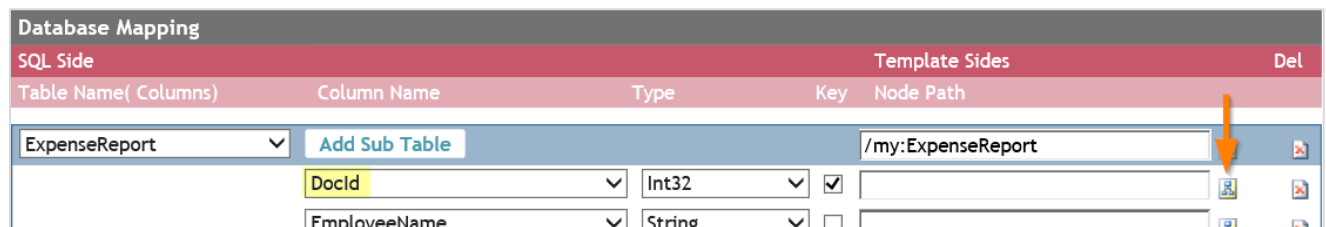
11. Click the **Select Schema Node** icon.

Database Mapping				
SQL Side			Template Sides	
Table Name(Columns)	Column Name	Type	Key	Node Path
ExpenseReport				
Add Sub Table DocId EmployeeName Purpose ReportDate Int32 String String Select... ☒ ☐ ☐ ☐ 				

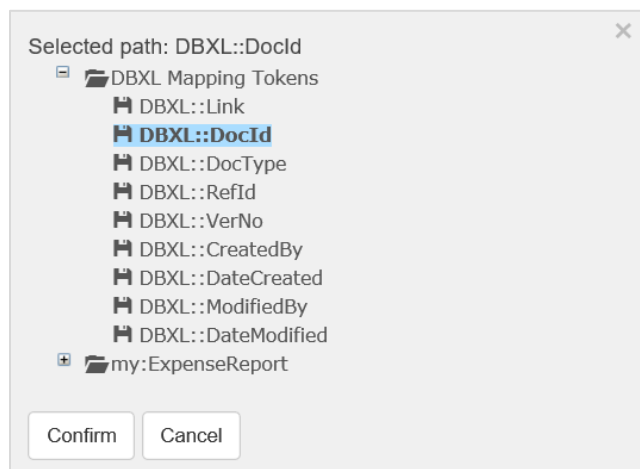
12. In the task pane, select the root node from the form and click **Confirm**.



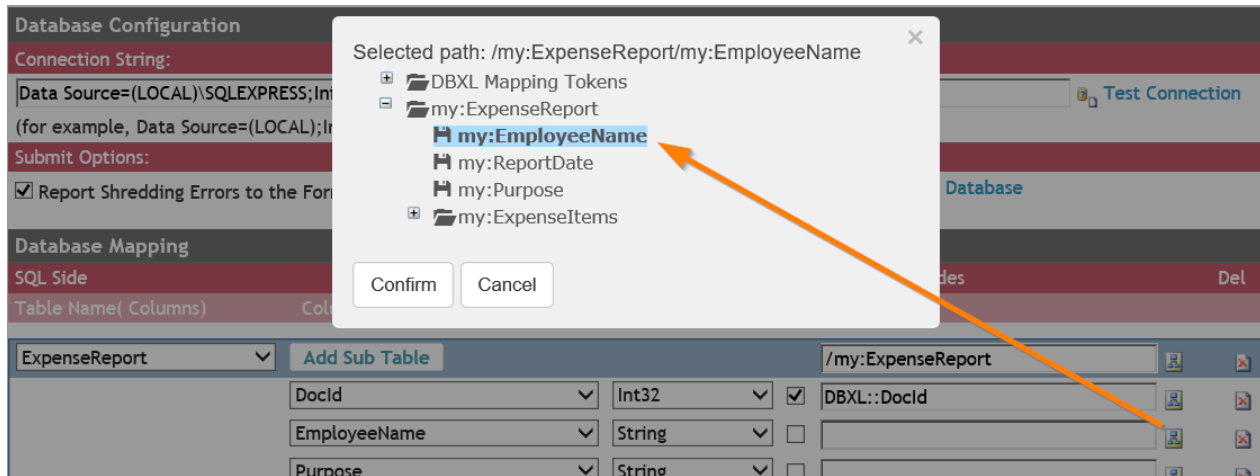
13. Select the **Select Schema Node** icon for the **DocId** column.



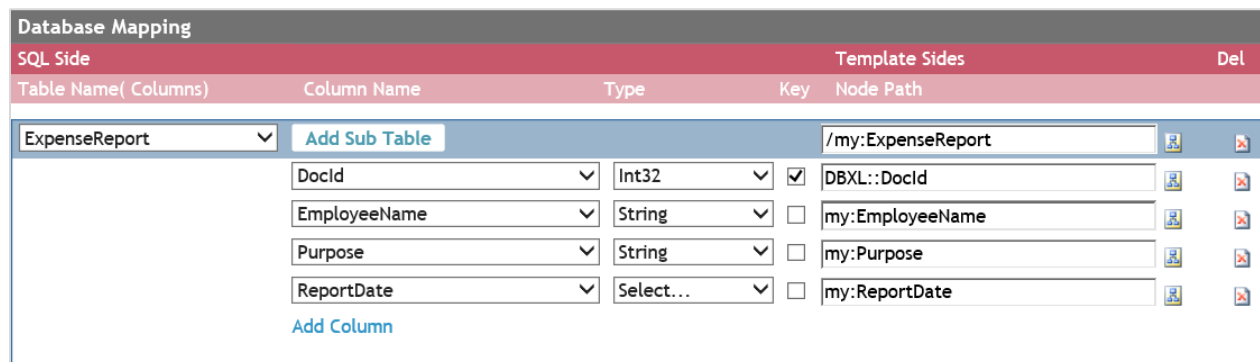
14. In the task pane, select **DBXL::DocId** under **DBXL Mapping Tokens** node. Click **Confirm**.



15. For the rest of the columns, click the **Select Schema Node** icon of each of the columns and select the matching field in the form node.



Here's the final setup:



To test the new database mapping, you can either do the following:

- Simply click **Reshred All Documents** in the **Database** tab if you have existing documents:

Database Configuration

Connection String:
 Data Source=(LOCAL)\SQLEXPRESS;Initial Catalog=Demos;Integrated Security=True
 (for example, Data Source=(LOCAL);Initial Catalog=TestDB;Integrated Security=True) [Test Connection](#)

Submit Options: ☒ Report Shredding Errors to the Form User

Database Actions: [Reshred All Documents](#) [Refresh Database](#)

Database Mapping

SQL Side	Column Name	Type	Key	Template Sides	Del
Table Name(Columns)				Node Path	
ExpenseReport				/my:ExpenseReport	
	DocId	Int32	☒	DBXL::DocId	
	EmployeeName	String	☐	my:EmployeeName	
	Purpose	String	☐	my:Purpose	
	ReportDate	Select...	☐	my:ReportDate	

- b. Submit new ones if no documents exist. To do this, fill out the form and submit the data.

Expense Report

Employee Name:

Report Date:

Purpose:

Command	Result	Success	Error
SubmitToDbxl /submit=SubmitDocument ...	46910	☒ Success	

Use SQL Server Management Studio to verify that the new document has been mapped to your table:

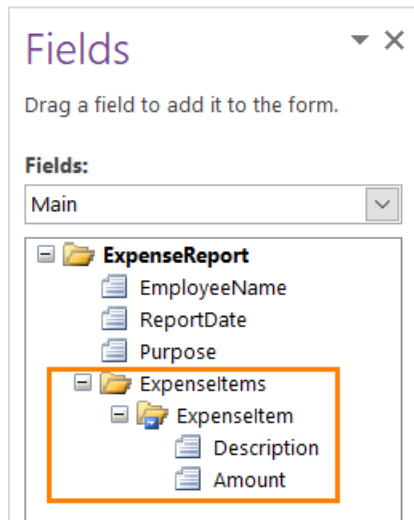
SELECT * FROM [Demos].[dbo].[ExpenseReport]

	EmployeeName	ReportDate	Purpose	DocId
1	tester	2019-01-07	testing	46910

Add Database Mapping for Nested/Repeating Group

Here's how you can map in DBXL the repeating data from your form to the SQL database.

1. Identify the repeating nodes in your form. In our example, we have the following fields under the repeating group **ExpenseItem**:



2. In SQL Server Management Studio, create a new table and add new columns that match these repeating nodes in your form. Remember: You can add as many or as few columns depending on the data that needs to be saved in the database.

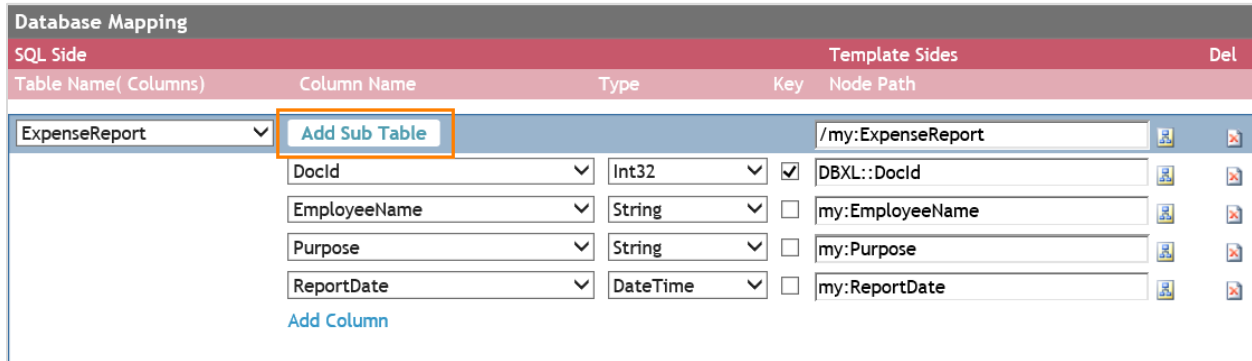
Column Name	Data Type	Allow Nulls
ExpenseReportId	int	☐
Description	nvarchar(50)	☒
Amount	decimal(18, 0)	☒

Note:

In our example, we added an additional column, **ExpenseReportId**, because this is what links the child data (items) to its parent table.

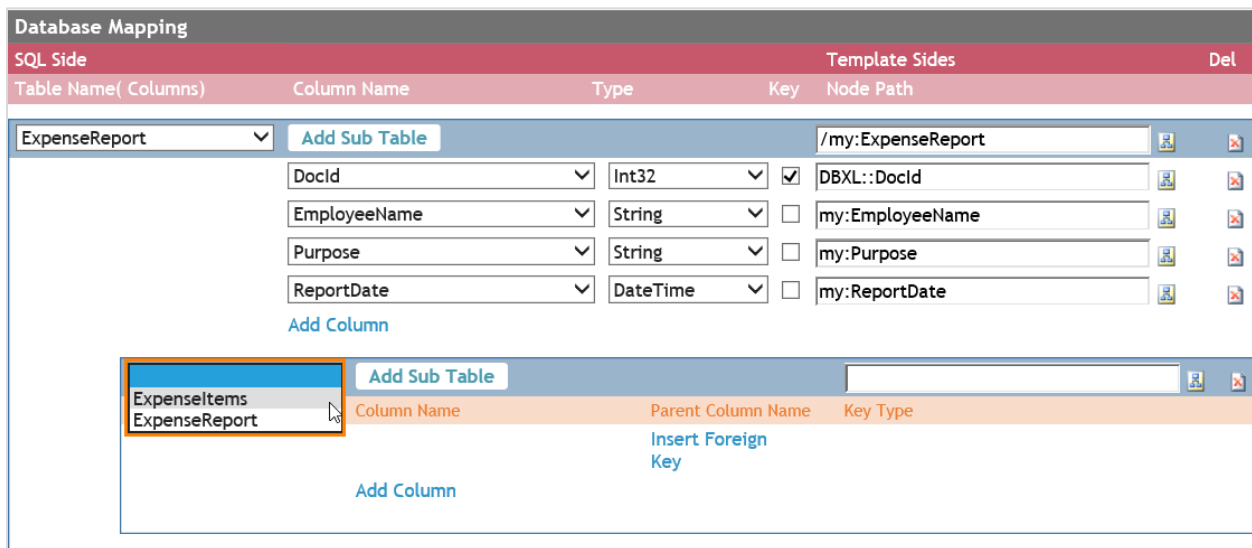
Add Mapping for the Child Table

1. Back to the DAT, select the DocType and switch to **Database** tab. Then Click **Add Sub Table**.



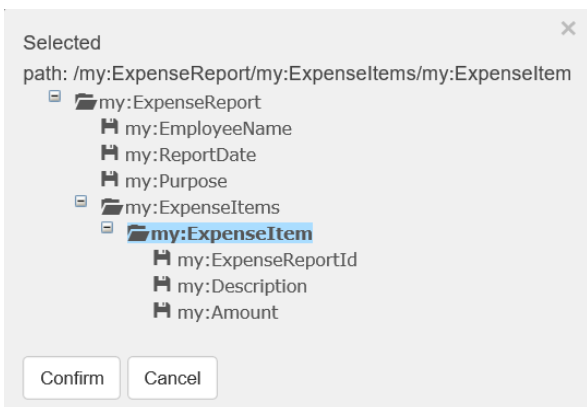
Database Mapping					
SQL Side			Template Sides		Del
Table Name(Columns)	Column Name	Type	Key	Node Path	
ExpenseReport				/my:ExpenseReport	
	DocId	Int32	☒	DBXL::DocId	
	EmployeeName	String	☐	my:EmployeeName	
	Purpose	String	☐	my:Purpose	
	ReportDate	DateTime	☐	my:ReportDate	
Add Column					

2. Select the child table from the dropdown.



Database Mapping					
SQL Side			Template Sides		Del
Table Name(Columns)	Column Name	Type	Key	Node Path	
ExpenseReport				/my:ExpenseReport	
	DocId	Int32	☒	DBXL::DocId	
	EmployeeName	String	☐	my:EmployeeName	
	Purpose	String	☐	my:Purpose	
	ReportDate	DateTime	☐	my:ReportDate	
Add Column					
ExpenseItems					
ExpenseReport					
Add Column					

3. Select the repeating group node by clicking the **Select Schema Node** icon.



Selected

path: /my:ExpenseReport/my:ExpenseItems/my:ExpenseItem

- my:ExpenseReport
 - my:EmployeeName
 - my:ReportDate
 - my:Purpose
 - my:ExpenseItems
 - my:ExpenseItem**
 - my:ExpenseReportId
 - my:Description
 - my:Amount

Confirm Cancel

Database Mapping

SQL Side				Template Sides	Del
Table Name(Columns)	Column Name	Type	Key	Node Path	
ExpenseReport	Add Sub Table			/my:ExpenseReport	
	DocId	Int32	☒	DBXL::DocId	
	EmployeeName	String	☐	my:EmployeeName	
	Purpose	String	☐	my:Purpose	
	ReportDate	DateTime	☐	my:ReportDate	
Add Column					
Expenseltems	Add Sub Table			/my:ExpenseReport/my:Expenseltems/	
Foreign Key: Column Name		Parent Column Name	Key Type		
		Insert Foreign Key			
Add Column					

- Click **Add Column** and select all columns from the child table and select the corresponding nodes in the form by clicking the **Select Schema Node** icon.

Database Mapping

SQL Side				Template Sides	Del
Table Name(Columns)	Column Name	Type	Key	Node Path	
ExpenseReport	Add Sub Table			/my:ExpenseReport	
	DocId	Int32	☒	DBXL::DocId	
	EmployeeName	String	☐	my:EmployeeName	
	Purpose	String	☐	my:Purpose	
	ReportDate	DateTime	☐	my:ReportDate	
Add Column					
Expenseltems	Add Sub Table			/my:ExpenseReport/my:Expenseltems/	
Foreign Key: Column Name		Parent Column Name	Key Type		
		Insert Foreign Key			
	Amount	Decimal	☐	my:Amount	
	Description	String	☐	my:Description	
Add Column					

- To link the two tables together, specify the foreign key by clicking **Insert Foreign Key**.
- Select the child table Id in the dropdown and for the **Parent Column Name**, choose **DocId** in the dropdown.

Database Mapping

SQL Side				Template Sides		Del
Table Name(Columns)	Column Name	Type	Key	Node Path		
ExpenseReport	Add Sub Table			/my:ExpenseReport		
	DocId	Int32	☒	DBXL::DocId		
	EmployeeName	String	☐	my:EmployeeName		
	Purpose	String	☐	my:Purpose		
	ReportDate	DateTime	☐	my:ReportDate		
Add Column						
ExpenseItems	Add Sub Table			/my:ExpenseReport/my:ExpenseItems/		
Foreign Key: Column Name Parent Column Name Key Type						
	ExpenseReportId	DocId	Column			
Insert Foreign Key						
	Amount	Decimal	☐	my:Amount		
	Description	String	☐	my:Description		
Add Column						

- Click **Save**, and then click **OK** in the confirmation dialog. Now let's verify the mapping!
- To test the mapping, you can submit a new form or reshred existing documents.

```
SELECT * FROM [Demos].[dbo].[ExpenseItems]
```

100 % <

Results Messages

	Description	Amount	ExpenseReportId
1	test	21	46913

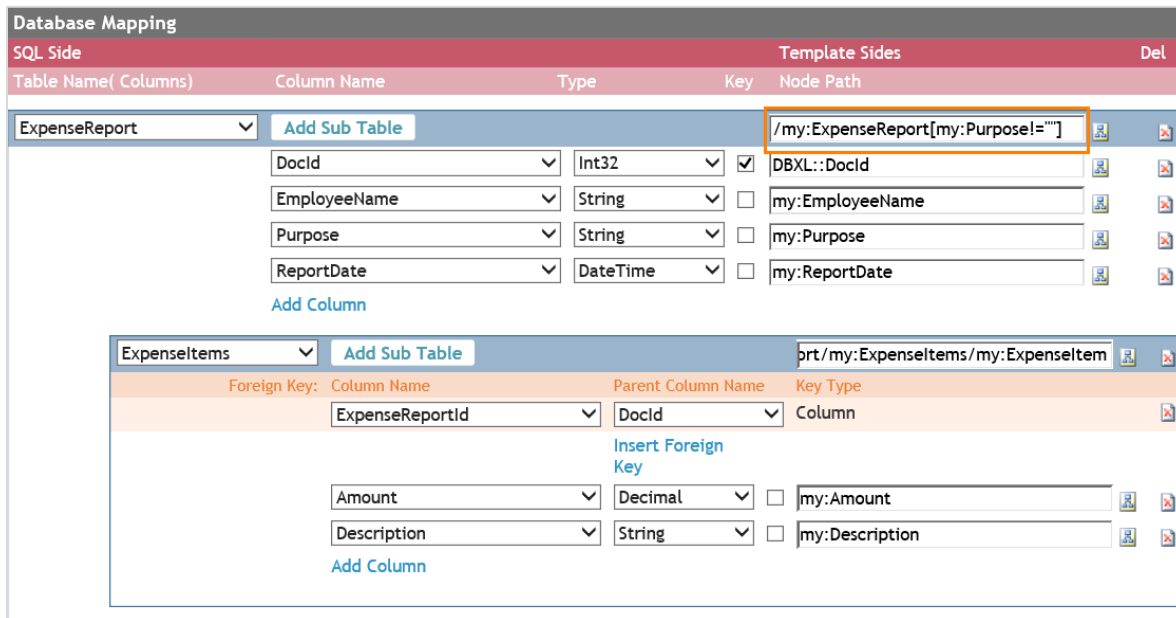
Conditional Shredding

As we have seen, DBXL allows us to map information from the xml documents into SQL tables. There are instances where we do not want all mapped data to propagate to the tables, let's say, fields containing NULLs. It turns out that one can leverage the power of XPath to create conditions on when shredding can occur.

To implement this, we only need one small change to our existing database mapping.

1. In **DAT**, select your DocType and switch to **Database** tab.
2. Modify the **Node Path** for the parent table.
3. Suppose we do not want to gather information from documents having blank **Purpose** field. So, we add the following in the **Node Path**, which basically means to only reshred data when the **Purpose** field is populated.

`/my:ExpenseReport[my:Purpose!=""]`



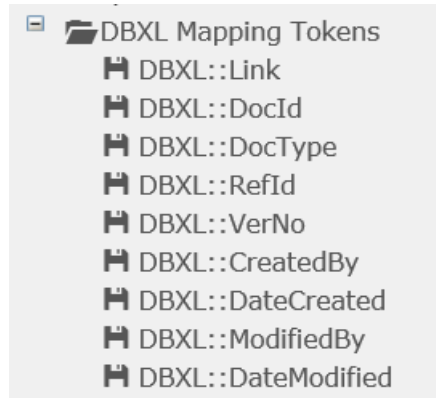
The screenshot shows the 'Database Mapping' interface. The top section is for the 'ExpenseReport' table, which is selected in the 'Table Name' dropdown. The 'Node Path' field is highlighted with an orange box and contains the XPath expression `/my:ExpenseReport[my:Purpose!='']`. Below this, the table's columns are listed: 'DocId' (Int32, Key), 'EmployeeName' (String), 'Purpose' (String), and 'ReportDate' (DateTime). The bottom section shows the 'ExpenseItems' table, which is configured as a foreign key to 'ExpenseReport'. The 'Foreign Key' is 'ExpenseReportId', the 'Parent Column Name' is 'DocId', and the 'Key Type' is 'Column'. The 'ExpenseItems' table has columns 'Amount' (Decimal) and 'Description' (String).

4. Click **Save** and then click **OK** in the confirmation dialog. To verify this change, simply submit a new form, but without any information for the **Purpose** field.

This guarantees that incomplete data will not be mapped to SQL.

Mapping DBXL Tokens

When creating your database mapping in DAT, you might have noticed nodes under a folder called **DBXL Mapping Tokens**.



This group contains the document's link, its DocID, the DocType it belongs to, its RefID and its version number. It also contains change information such as the Created/Modified By and Creation/Modified data. In our example above, we only used the DocID, but you can use any of these tokens to map data into your SQL database.