

MAP YOUR FORM DATA INTO SQL

Using DBXL you can create mappings from your XML forms to your SQL tables.

In order to execute the steps in this document, you will need an InfoPath form that submits to DBXL. Luckily, we have already done this in the first module.

CREATE A SQL TABLE TO STORE THE DATA

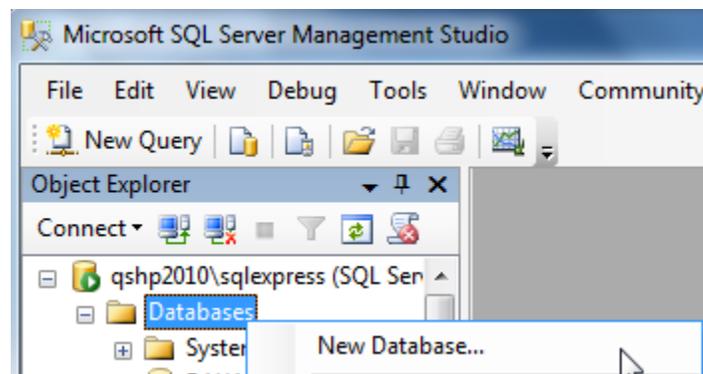
These steps will be demonstrated by the presenter.

First, you will need to create a SQL database. This exercise will initially require one table.

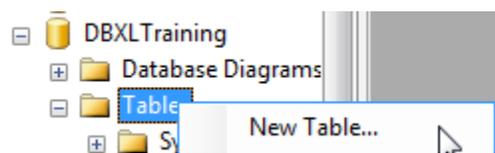
1. Open SQL Server Management Studio Express.
2. Connect to the SQL instance. For the purposes of this training module, we recommend using the same database that was used to install the QdabraDBXL and QdabraUtility databases.

Note that we don't recommend adding tables to the same databases (**QdabraDBXL** and **QdabraUtility**) that DBXL uses.

3. Create a new database and call it DBXLTraining.

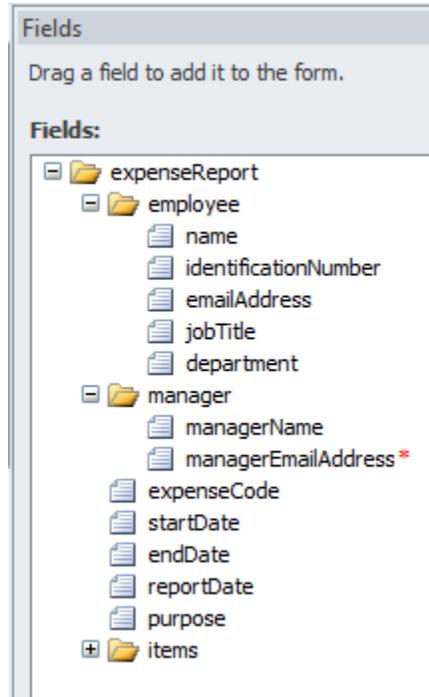


4. Right click on that database, select **New Table**.



Qdabra DBXL Training: Module 2

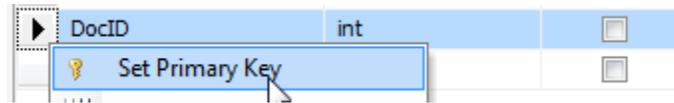
Recall your InfoPath form template's schema:



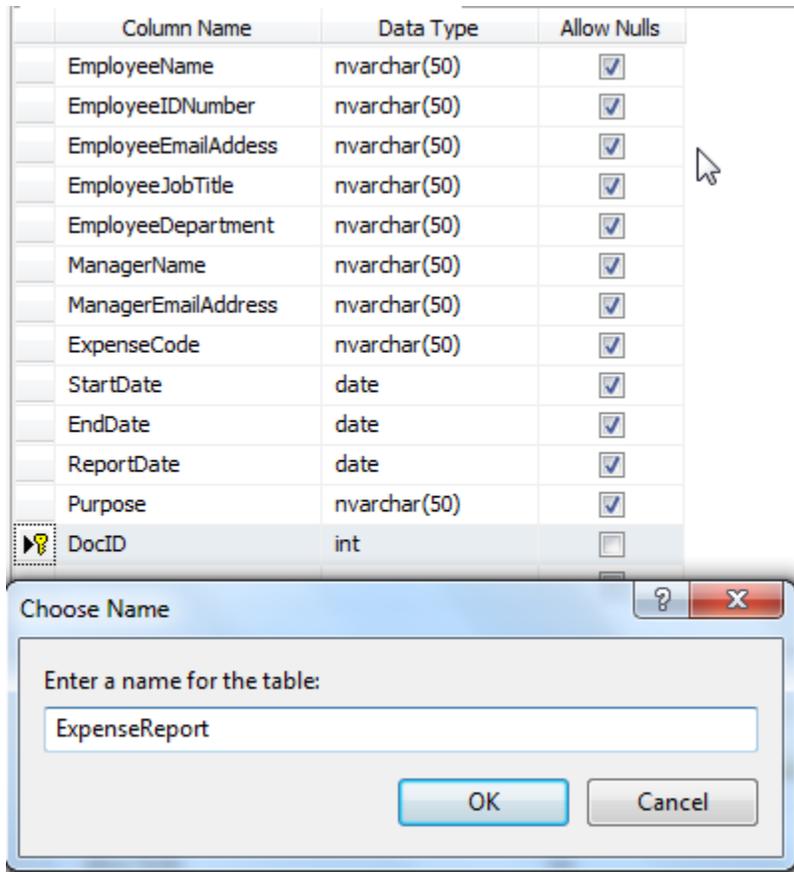
5. Add columns to the SQL table, as shown below:

Column Name	Data Type
EmployeeName	nvarchar(50)
EmployeeIDNumber	nvarchar(50)
EmployeeEmailAddress	nvarchar(50)
EmployeeJobTitle	nvarchar(50)
EmployeeDepartment	nvarchar(50)
ManagerName	nvarchar(50)
ManagerEmailAddress	nvarchar(50)
ExpenseCode	nvarchar(50)
StartDate	date
EndDate	date
ReportDate	date
Purpose	nvarchar(50)

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



7. Save the table as **ExpenseReport**.



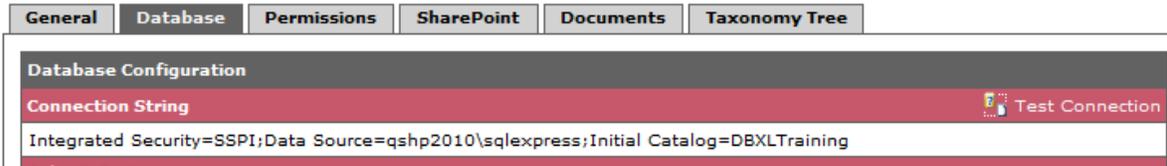
CONFIGURE A SIMPLE SQL MAPPING IN DAT

In the following steps, you will be instructed to setup 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.

8. Open the DBXL Administration Tool.
9. Click **Edit** for the Document Type.
10. In DAT, click on the **Database** tab.
11. Enter the Data Connection String appropriate for your SQL server. For example, ***Integrated Security=SSPI;Data Source=(LOCAL)\sqlexpress;Initial Catalog=DBXLTraining***

Qdabra DBXL Training: Module 2

12. Tab out of the Connection String field.
13. Click **Test Connection**, and click **OK** when success is confirmed.



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

14. Click **Save**, and then click **OK** in the confirmation dialog.
15. Click **Add Table**.
16. Click on the **Select Database Table** icon () found next to the **Table Name** field. This will load the SQL schema in the Task Pane.
17. In the taskpane, double click on the name of the table.
18. Click **Add Column**.
19. Click on the **Select Database Column** icon () found next to the **Column Name** field. This will load the SQL schema in the Task Pane.
20. Double click on the first column, **EmployeeDepartment**.
21. Repeat the previous three steps to add the other table columns.

Qdabra DBXL Training: Module 2

Database Map	
Table Name (12 Columns) ▼	M
ExpenseReport	 /
Column Name	
EmployeeDepartment	
EmployeeEmailAddress	
EmployeeIDNumber	
EmployeeJobTitle	
EmployeeName	
EndDate	
ExpenseCode	
ManagerEmailAddress	
ManagerName	
Purpose	
ReportDate	
StartDate	

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.

22. Click on the **Select Schema Node** icon () under **Node Path**, for the table.
23. In the taskpane, double click on the my:expenseReport node.
24. Click on the **Select Schema Node** icon () under **Node Path**, for the **EmployeeDepartment** column.
25. In the taskpane, double click on the **my:employee/my:department** node.
26. Repeat the previous two steps for the rest of the rows in the mapping, always using the taskpane to select the form node.

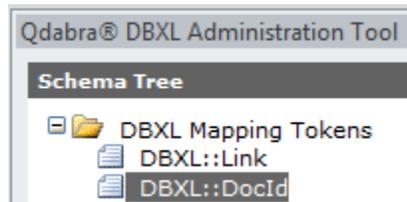
Your mapping should look like this:

Qdabra DBXL Training: Module 2

Table Name (13 Columns) ▼		Node Path		
ExpenseReport		/my:expenseReport		
Column Name	DB Type	Key	Node Path	
EmployeeDepartment	String	<input type="checkbox"/>	my:employee/my:department	
EmployeeEmailAddress	String	<input type="checkbox"/>	my:employee/my:emailAddress	
EmployeeIDNumber	String	<input type="checkbox"/>	my:employee/my:identificationNumber	
EmployeeJobTitle	String	<input type="checkbox"/>	my:employee/my:jobTitle	
EmployeeName	String	<input type="checkbox"/>	my:employee/my:name	
EndDate	DateTime	<input type="checkbox"/>	my:endDate	
ExpenseCode	String	<input type="checkbox"/>	my:expenseCode	
ManagerEmailAddress	String	<input type="checkbox"/>	my:manager/my:managerEmailAddress	
ManagerName	String	<input type="checkbox"/>	my:manager/my:managerName	
Purpose	String	<input type="checkbox"/>	my:purpose	
ReportDate	DateTime	<input type="checkbox"/>	my:reportDate	
StartDate	DateTime	<input type="checkbox"/>	my:startDate	
DocID	Int32	<input checked="" type="checkbox"/>		

What about the DocID?

- Click on the **Select Schema Node** icon () under **Node Path**, for the **DocID** column.
- In the taskpane, double click on the **DBXL::DocId** node.



- Click **Save**, and then click **OK** in the confirmation dialog.

Let's find out if the mapping is working.

VERIFY MAPPING

- Click on the **General** tab.
- Click **Open**.
- Fill out the form. When done, click the Submit button.
- Click the **Documents** tab in DAT. The new document is added to the **Documents** table.
- Use SQL Server Management Studio Express to verify that the new document has been mapped to your table. **This step will be demonstrated by the presenter.**

Qdabra DBXL Training: Module 2

```
SELECT *
FROM [DBXLTraining].[dbo].[ExpenseReport]
```

	EmployeeName	EmployeeIDNumber	EmployeeEmailAddress	EmployeeJobTitle	EmployeeDepartment	ManagerName	ManagerEmailAddress
1	John Smith	1234	john.smith@qdabra.com	Technician	IT	Bill Smith	bill.smith@qdabra.com

In lab 1 we submitted a document, but this was BEFORE creating the database mapping, follow these steps to map that data into your table.

35. Click Reshred All Documents and click YES on the warning dialog.

Doc. ID	Ref. ID	Version #	Name	Author	Description	New	
6461	1	1	2012-03-18	Ernesto Machado	This is a test document.	Open	<input checked="" type="checkbox"/>
6462	2	1	2012-03-18	John Smith	This is my business purpose!	Open	<input type="checkbox"/>

36. Click OK in the confirmation dialog.

37. Use SQL Server Management Studio Express to verify that the data was mapped to your table. **This step will be demonstrated by the presenter.**

```
SELECT *
FROM [DBXLTraining].[dbo].[ExpenseReport]
```

	EmployeeName	EmployeeIDNumber	EmployeeEmailAddress	EmployeeJobTitle	EmployeeDepartment	ManagerName	ManagerEmailAddress
1	Ernesto Machado	NULL	NULL	NULL	NULL	NULL	patrick.halstead@qdabra.com
2	John Smith	1234	john.smith@qdabra.com	Technician	IT	Bill Smith	bill.smith@qdabra.com

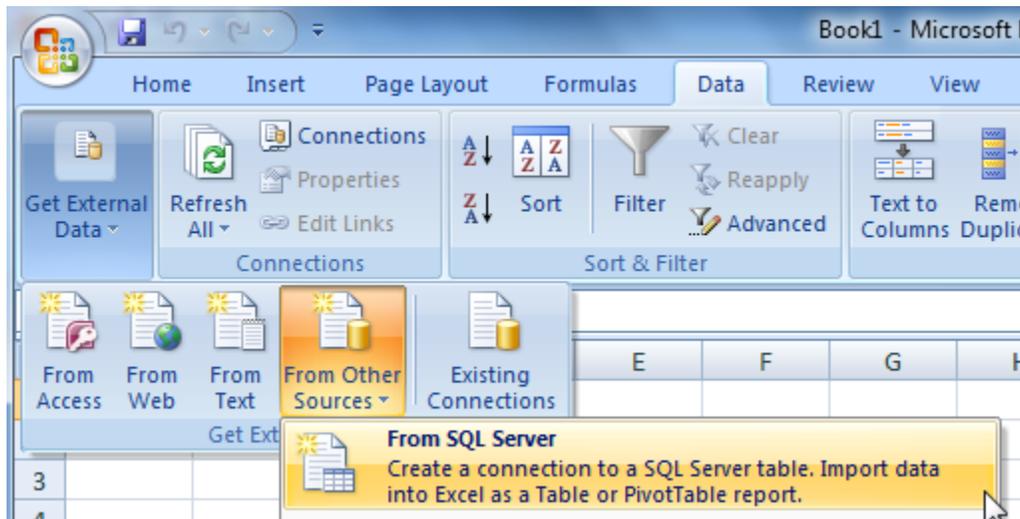
The mapping between DBXL and SQL has now been created and verified!

CREATE AN EXCEL REPORT

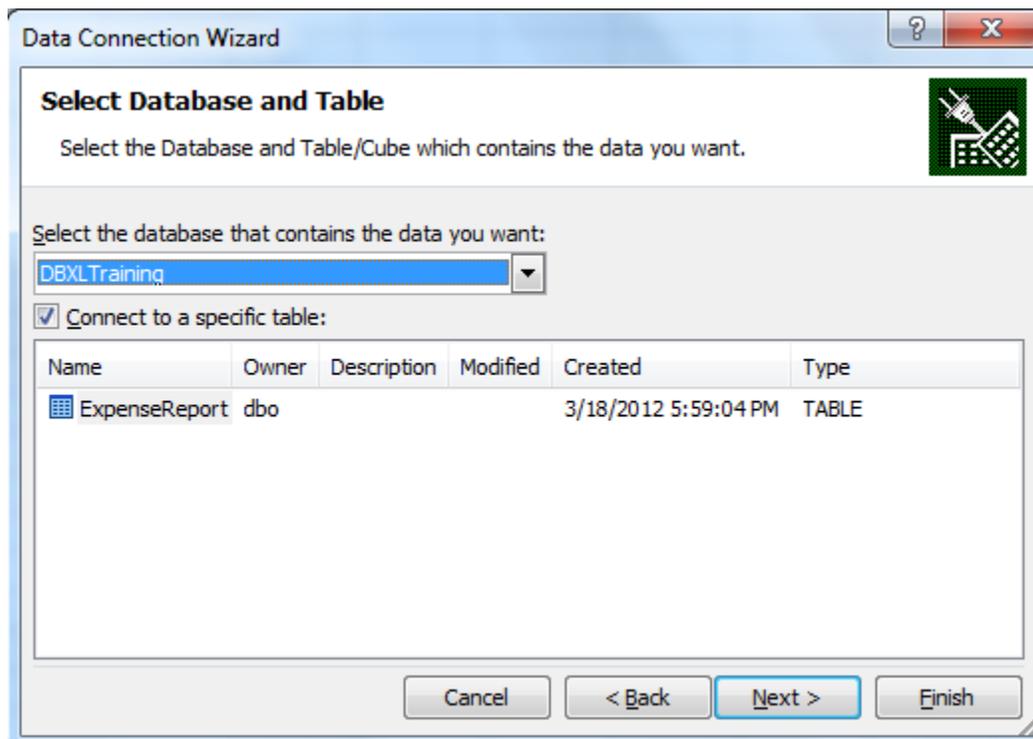
38. Launch Microsoft Excel.

39. Select **Data > Get External Data > From Other Sources > From SQL Server.**

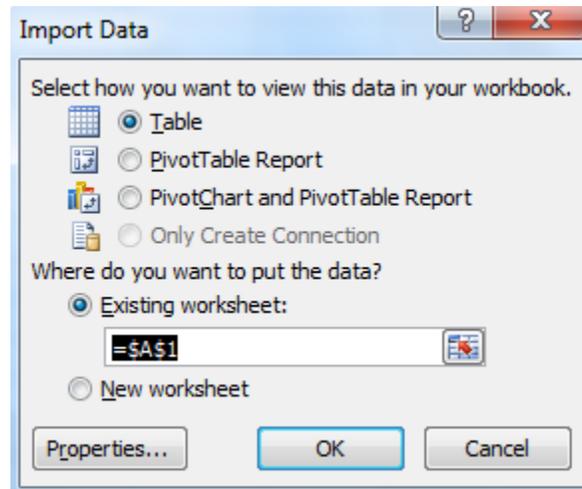
Qdabra DBXL Training: Module 2



40. Enter the SQL instance name and click **Next**.
41. Select the correct database and table, and click **Finish**.



42. Click OK in the Import Data dialog. Though there are various options in this dialog, we'll just accept the defaults.



Excel now creates a table in the worksheet, showing the data that DBXL mapped to SQL.

	A	B	C	D	E	F
1	EmployeeName	EmployeeIDNumber	EmployeeEmailAddress	EmployeeJobTitle	EmployeeDepartment	ManagerName
2	Ernesto Machado					
3	John Smith	1234	john.smith@qdabra.com	Technician	IT	Bill Smith

43. Go back to the DBXL Administration Tool, switch to the **General** tab, and click **Open**.
44. Fill out and submit a new document.
45. Back in Excel, in the **Data** tab, click **Refresh All**.

The new record is now displayed in your Excel report:

	A	B	C	D	E	F
1	EmployeeName	EmployeeIDNumber	EmployeeEmailAddress	EmployeeJobTitle	EmployeeDepartment	ManagerName
2	Ernesto Machado					
3	John Smith	1234	john.smith@qdabra.com	Technician	IT	Bill Smith
4	Jane Doe	97391	jane.doe@qdabra.com	Project Manager	Sales	Bill Smith

46. Close Excel. Save the report if you'd like.

This is the simplest possible example, where the data is flat. But what happens if the data is not flat? The next section will walk you through modifying the form such that it has a parent-child relationship. Then, you will modify the database mapping to reflect this new structure.

ADD A CHILD TABLE IN SQL

These steps will be demonstrated by the presenter.

Now we want to map the repeating data for Items, and to do that we need an additional SQL table.

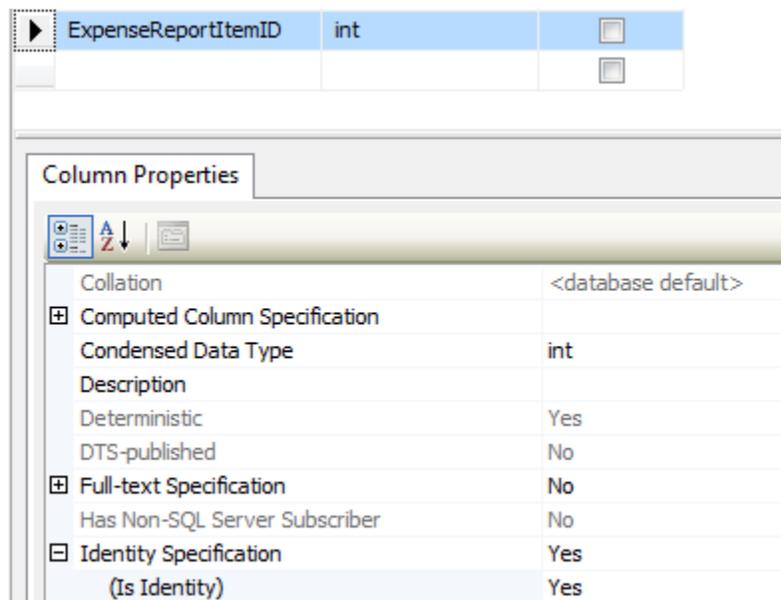
Qdabra DBXL Training: Module 2

47. In SQL Server Management Studio, create a new table and call it ExpenseReportItems. It will have the columns shown in the screenshot below.

Column Name	Data Type
ExpenseReportId	bigint
Date	datetime
Description	nvarchar(128)
Amount	decimal(18, 2)
Category	nvarchar(50)
MerchantName	nvarchar(128)
EntertainmentPurpose	nvarchar(512)
EntertainmentPlace	nvarchar(100)
AdditionalInfo	nvarchar(MAX)

Why do we need the ExpenseReportId in the child table? Because this is what links our child (Items) data to its parent table.

48. Add an additional column called ExpenseReportItemID.
 49. Set this column to be the Primary key, and also set it to auto-increment (Identity Specification = Yes).



ADD MAPPING FOR THE CHILD TABLE

Qdabra DBXL Training: Module 2

50. Back in the DBXL Administration Tool, click **Edit** for the document type.
51. Switch to the **Database** tab.
52. Click on **Add Sub Table**.
53. Click on the **Select Database Table** icon () found next to the **Table Name** field.
54. In the taskpane, double click on the node for the **ExpenseReportItems** table.
55. Click **Add Column**, directly under the new row in the mapping.
56. Click on the **Select Database Column** icon () found next to the **Column Name** field. This will load the SQL schema in the Task Pane. Double click on the **AdditionalInfo**. Repeat the for the other table columns.

Table Name (8 Columns) ▼		Ne
ExpenseReportItem 		m
Column Name		
<input checked="" type="checkbox"/> Insert Foreign Key		
Column Name		
AdditionalInfo		
Amount		
Category		
Date		
Description		
EntertainmentPlace		
EntertainmentPurpose		
MerchantName		

57. Click on the **Select Schema Node** icon under **Node Path**, for the **ExpenseReportItem** table.
58. In the taskpane, double click on the **my:items/my:item** node.
59. Click on the **Select Schema Node** icon under **Node Path**, for the **AdditionalInfo** column.
60. In the taskpane, double click on the **my:additionalInfo** node.
61. Repeat the previous two steps for the other rows.

Qdabra DBXL Training: Module 2

Table Name (8 Columns) ▼		Node Path	
ExpenseReportItem		my:items/my:item	
Column Name	Key Type	Parent Column Name	
Insert Foreign Key			
Column Name	DB Type	Key	Node Path
AdditionalInfo	String	<input type="checkbox"/>	my:additionalInfo
Amount	Decimal	<input type="checkbox"/>	my:amount
Category	String	<input type="checkbox"/>	my:category
Date	DateTime	<input type="checkbox"/>	my:date
Description	String	<input type="checkbox"/>	my:description
EntertainmentPlace	String	<input type="checkbox"/>	my:entertainmentDetails/my:entertainmentPlace
EntertainmentPurpose	String	<input type="checkbox"/>	my:entertainmentDetails/my:entertainmentPurpos
MerchantName	String	<input type="checkbox"/>	my:merchantName

Now you will link the two tables together.

62. Click on **Insert Foreign Key**.
63. Click on the **Select Database Column** icon.
64. Double click on **ExpenseReportId** in the taskpane.
65. Under Parent Column Name, click on the **Select Database Column** icon.
66. In the taskpane, double click on the **DocID** node for the **ExpenseReport** table.

Your mapping should look like this:

Column Name	Key Type	Parent Column Name
ExpenseReportId	Column	DocID

You might be left wondering why we did not map the ExpenseReportItemID column. It's because we set that be an autoincrement, primary key when we created the table. Therefore, there is no need to map a value from the InfoPath form.

67. Click **Save**, and then click **OK** in the confirmation dialog.

Now let's verify the mapping!

68. Click **Reshred All Documents**, and then click OK in the confirmation message.
69. Use SQL Server Management Studio Express to verify that the item information has been mapped to your table.

```
SELECT *
FROM [DBXLTraining].[dbo].[ExpenseReportItem]
```

	ExpenseReportId	Date	Description	Amount	Category
1	6461	2012-03-02 00:00:00.000	Hotel	100.00	Lodging
2	6461	2012-03-03 00:00:00.000	Dinner	50.00	Meals
3	6462	2012-03-01 00:00:00.000	Lunch	10.00	Meals
4	6462	2012-03-16 00:00:00.000	Dinner	20.00	Meals
5	6463	2012-03-08 00:00:00.000	New computer	1000.00	Office supplies

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.

70. In the DBXL Administration Tool, switch to the Database tab.

71. Modify the Node Path for the ExpenseReport table:

Table Name (13 Columns) ◀	Node Path
ExpenseReport	/my:expenseReport[my:manager/my:managerName !='']

The Node Path is now: ***/my:expenseReport[my:manager/my:managerName !='']***

72. Modify the Node Path for the ExpenseReportItem table:

Table Name (8 Columns) ▼	Node Path
ExpenseReportItem	my:items/my:item[my:description !='']

The Node Path is now: ***my:items/my:item[my:description !='']***

73. Click **Save**, and then click **OK** in the confirmation dialog.

74. To verify this change, simply submit a new form, but without any information for items.

Qdabra DBXL Training: Module 2

This guarantees that incomplete data will not be mapped to SQL and used for reporting.

```

SELECT *
FROM [DBXLTraining].[dbo].[ExpenseReport]
SELECT *
FROM [DBXLTraining].[dbo].[ExpenseReportItem]

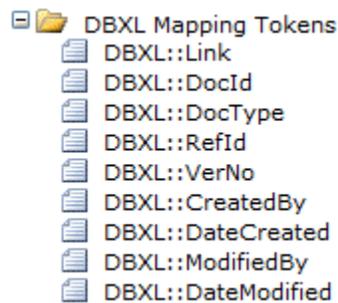
```

	EmployeeName	EmployeeIDNumber	EmployeeEmailAddress	EmployeeJob Title	EmployeeDepartment
1	John Smith	1234	john.smith@qdabra.com	Technician	IT
2	Jane Doe	97391	jane.doe@qdabra.com	Project Manager	Sales

	ExpenseReportId	Date	Description	Amount	Category	MerchantName
1	6462	2012-03-01 00:00:00.000	Lunch	10.00	Meals	NULL
2	6462	2012-03-16 00:00:00.000	Dinner	20.00	Meals	NULL
3	6463	2012-03-08 00:00:00.000	New computer	1000.00	Office supplies	NULL

MAPPING DBXL TOKENS

When creating your database mapping in the DBXL Administration Tool (DAT) you might have noticed nodes under a folder called "DBXL Mapping Tokens".



That group contains (for example) the document's link, its DocID, the DocType it belongs to, its RefID and its version number.

In our example above, we only used the DocID., but you can use any of these tokens to map data into your SQL database, as seen below.

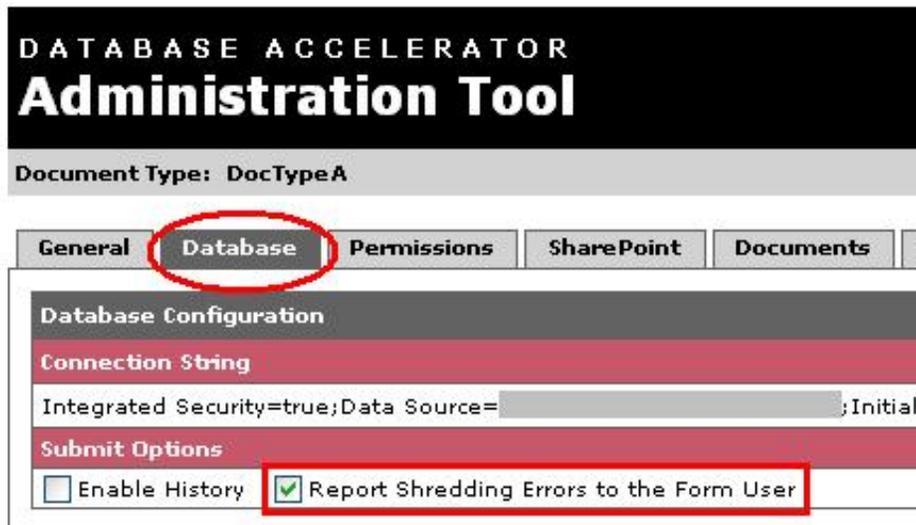
Qdabra DBXL Training: Module 2

Link	 String	<input type="checkbox"/>	DBXL::Link	
DocID	 Int32	<input type="checkbox"/>	DBXL::DocId	
DocType	 String	<input type="checkbox"/>	DBXL::DocType	
RefId	 Int32	<input type="checkbox"/>	DBXL::RefId	
VerNo	 String	<input type="checkbox"/>	DBXL::VerNo	
CreatedBy	 String	<input type="checkbox"/>	DBXL::CreatedBy	
DateCreated	 String	<input type="checkbox"/>	DBXL::DateCreated	
ModifiedBy	 String	<input type="checkbox"/>	DBXL::ModifiedBy	
DateModified	 String	<input type="checkbox"/>	DBXL::DateModified	

HOW TO REPORT SHREDDING ERRORS TO FORM USERS

A DBXL user might encounter an error when submitting a form to the SQL database. If the InfoPath form lacks validation, for example, a user might enter text into a field that is mapped into a numeric column. The mapping between DBXL and SQL will fail and an error will be logged in the Event Viewer on the server where DBXL is installed.

In order to help you identify these errors, you can use DBXL’s “Report Shredding Errors to the Form User” function which can be found in the DBXL Administration Tool’s Database tab, as you can see in the screenshot below:

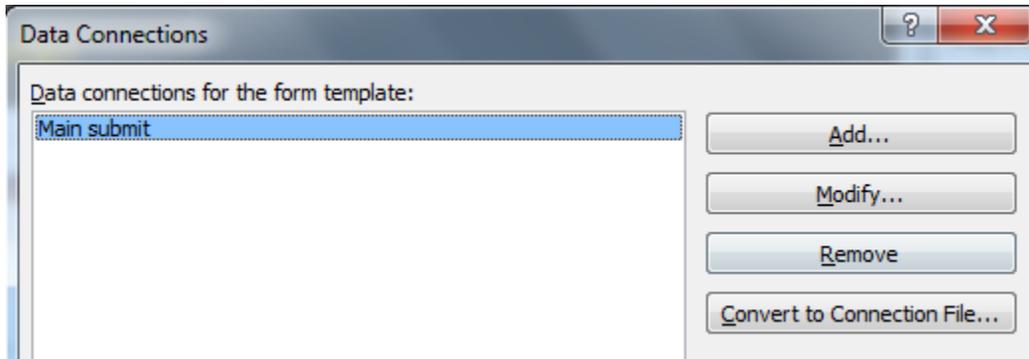


To use this function, your form should have a data connection that uses the DBXLDocumentServiceEx Web Service. Follow the steps below to modify the submit data connection in your form:

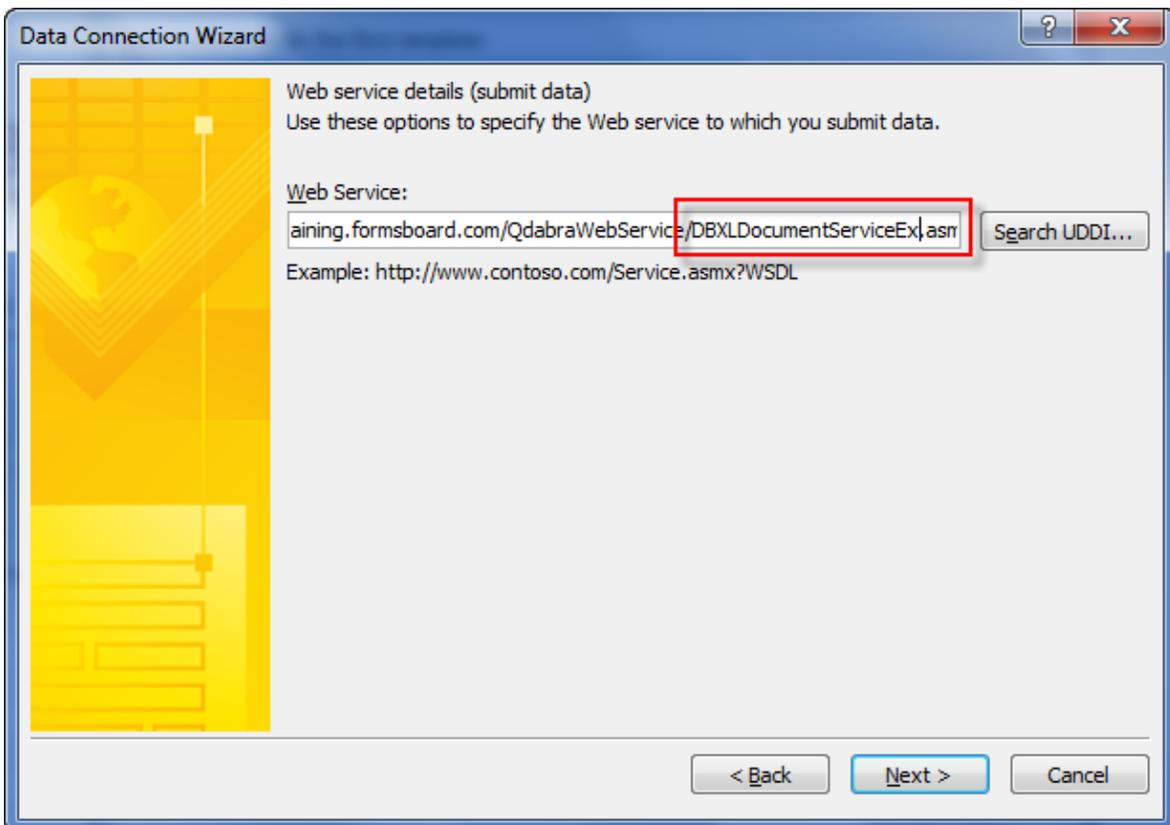
75. Right click on the InfoPath form and select **Design**.
76. Go to **Data > Data Connections**.

Qdabra DBXL Training: Module 2

77. Select the data connection that submits to DBXL (which was created in Module 1) and click **Modify**.



78. Change the web service URL from <http://<servername>/QdabraWebService/DbxlDocumentService.aspx> to <http://<servername>/QdabraWebService/DbxlDocumentServiceEx.aspx>.



79. In spite of this change, all other settings in the data connection should be preserved. Click **Next** three times, then click **Finish** and then click **Close**.

Qdabra DBXL Training: Module 2

80. Save your form.

Now, configure DAT to use the updated form.

81. In DAT, attach the form as a DocType.

82. Click the **Save** button in DAT.

83. Go to the **Database** tab, and make sure the **Report Shredding Errors to Form User** check box is selected. If not, select it.

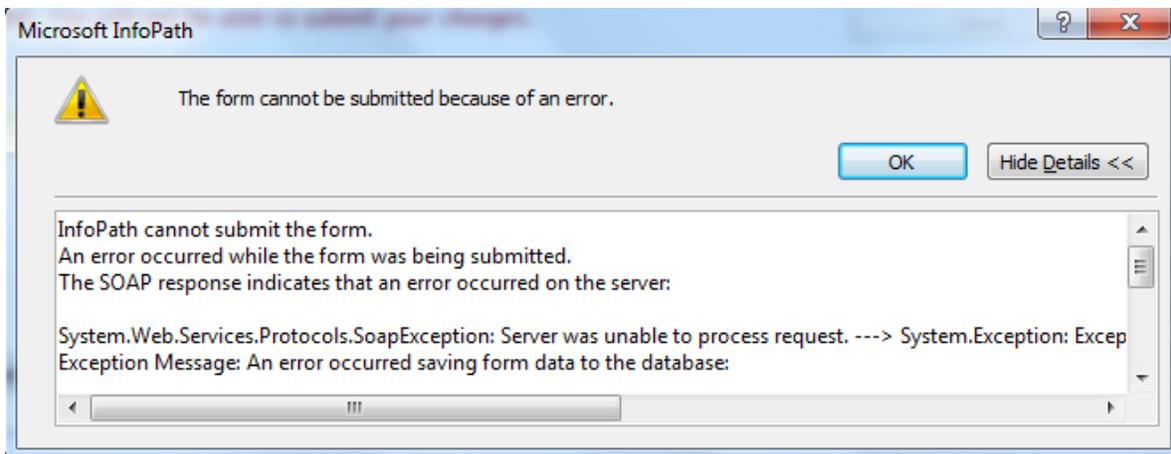


For the purpose of this training module, we will submit a form to incite an error. If we think back to the SQL table where the data is mapped, you'll remember that the columns were of data type nvarchar(50). This means that if we exceed 50 characters in any of the form field's we'll generate an error.

84. Click on the **Documents** tab.

85. Click the **New** link to create a new document.

86. Fill out the form but enter a **Purpose** that exceeds 50 characters and click **Submit**.



The error reads:

InfoPath cannot submit the form.

Qdabra DBXL Training: Module 2

*An error occurred while the form was being submitted.
The SOAP response indicates that an error occurred on the server:*

*System.Web.Services.Protocols.SoapException: Server was unable to process request. ---> System.Exception: Exception Code: W0001
Exception Message: An error occurred saving form data to the database:*

-One of the form's values was too large or could not be converted to the correct data type to be stored in the database.

As a result, the data has not been shredded to the database.

```
at
Qdabra.Dbxl.DbxlDocumentServiceEx.CheckAndThrowException(StatusInfo
info)
at
Qdabra.Dbxl.DbxlDocumentServiceEx.Qdabra.Dbxl.Services.IDbxlDocumen
tServiceEx.SubmitDocument(String docTypeName, String xml, String
name, String author, String description, Int32& docId, String&
refId)
--- End of inner exception stack trace ---
```