

## QUERY DATA FROM A SQL DATABASE

Using the QueryDB web service included in DBXL, we can set up our InfoPath forms to query data from SQL databases. This lab will walk you through an example.

---

### PRE-REQUISITE: ADD DATA TO SQL

1. Open SQL Server Management Studio and connect to your SQL server.
2. Execute the SQL script below to create the table in the DBXLTraining database.

```
USE [DBXLTraining]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[QueryDB_SourceData] (
    [Department] [nvarchar] (50) NULL,
    [Position] [nvarchar] (50) NULL,
    [ID] [int] IDENTITY (1,1) NOT NULL
) ON [PRIMARY]
GO
```

3. In the Object Explorer, right click on the top node and select **Refresh**.
4. Right click on the new table, called QueryDB\_SourceData, and select **Edit Top 200 Rows**.
5. Add the following data. We will query this data during this lab and use it in the InfoPath form.

Department	Position	ID
Support	Support engineer	1
Sales	Sales represent...	2
Support	Support manager	3
Sales	Sales manager	4
Administration	Accountant	5
Administration	HR	6

---

### PRE-REQUISITE: MODIFY FORM

For UI purposes, we will make a few changes to the canvas in the form.

Qdabra DBXL Training: Module 5

- Right click on the form template and select **Design**.

The Employee Information section currently looks like this:

Employee Information	
Name:	Title:
Department:	ID Number:
E-mail Address:	

- Reorganize the fields such that **Name** and **Email** are on the top row, and **Department** and **Title** in the middle row. **Id Number** won't be used in this lab, but place it in the third row in case it is needed later.

The final design will look like this:

Employee Information	
Name:	E-mail Address:
Department:	Title:
ID Number:	

## MODIFY DBXL'S WEB.CONFIG

We must make a small change in DBXL's web.config file to grant permissions to the web service and allows queries to this database.

These steps will be demonstrated by the presenter.

- Locate the web.config file (usually located in `C:\inetpub\wwwroot\QdabraWebService` on the server where DBXL was installed).
- Open the file in a text editor of your choice, such as Notepad.
- Locate the <databases> node.
- Enter a new line that contains the following: `<database alias="<Your_database>" name="<Your_database>" server="default" />`. In this case we are using

```
<database alias="DBXLTraining" name="DBXLTraining" server="default" />
```

## Qdabra DBXL Training: Module 5

In this case we are using the same value for the alias and the name, but as a DBXL administrator you could choose to make these different, thus “hiding” the name from the people designing the form template. The form designers will always refer to the database by its alias.

12. Save and close the web.config file.

---

### INSTALL QUERYBUILDER

Next, we will install the Query Builder tool which provides a special UI via an InfoPath form to allow us to create queries easily.

13. Access the Database Accelerator Control Panel from Internet Explorer; in default DBXL installation, the URL format is `http://<servername>/QdabraWebService/default.htm`.
14. On the page, scroll down to *Forms* and then click **Install Qdabra Query Builder**.



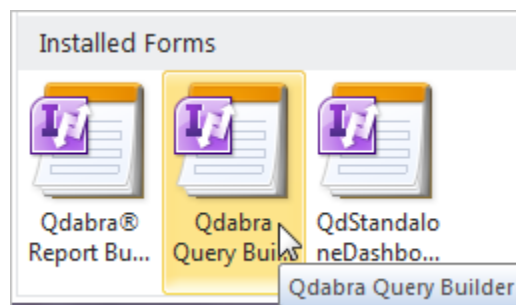
You will be prompted if you wish to run the installer or save it on your machine – run it and follow the on-screen instructions to install the tool.

---

### USE QUERYBUILDER TO GENERATE THE QUERYXML

Now that we have the tool installed, we can start building queries that we will use when we design our form later.

15. Open InfoPath Filler and select **Qdabra Query Builder** under *Installed Forms*.



16. When the form opens, start filling it out.
17. Enter the root URL where you installed DBXL:

Qdabra DBXL Training: Module 5

Data Sources		
Web Service URL Prefix (scheme://host-name:port)	Web Service Name	Target
http://<servername>	QdabraWebService	<input checked="" type="radio"/> SQL Database <input type="radio"/> SharePoint List

18. Leave the target as *SQL Database* and click **Construct Now**.

**Target**

SQL Database  SharePoint List

**Construct Now**

19. Another section will appear – fill out the *Data Source* and the *Initial Catalog* fields with your SQL server and your database name, respectively.

Database Connection String		Construct Now
Connection Parameters		
Data Source (Server Name)	Initial Catalog (Database Name)	Authentication
qshp2010\sqlexpress	DBXLTraining	<input checked="" type="radio"/> Windows <input type="radio"/> SQL Server
Data Source=qshp2010\sqlexpress;Initial Catalog=DBXLTraining;Integrated Security=True		

20. Click **Done**.

21. Under the *Query Parameters* section, click on the **Select Database Table** icon.

22. The *Database Schema Tree* task pane will appear on the right – double click on the QueryDB\_SourceData table to select it.

Query Parameters	
Database Name	Table Name
DBXLTraining	QueryDB_SourceData

Next we will specify the query columns.

23. Click **Insert return column** under *Query Columns*.

24. Click on the **Select Database Column** icon.

25. In the taskpane, double click on the Department node.

26. Click on **Build Now**.

27. Copy the second query string. It might look like this:

```
<query database='DBXLTraining' table='QueryDB_SourceData'><columns><column name='Department' /></columns></query>
```

28. Modify this query string such that it uses the alias that was defined in the web.config:

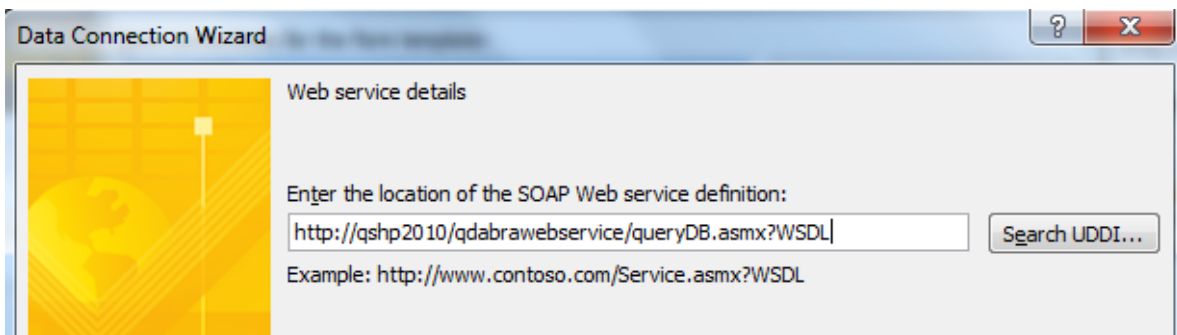
Qdabra DBXL Training: Module 5

```
<query database='#DBXLTraining#' table='QueryDB_SourceData'><columns><column name='Department' /></columns></query>
```

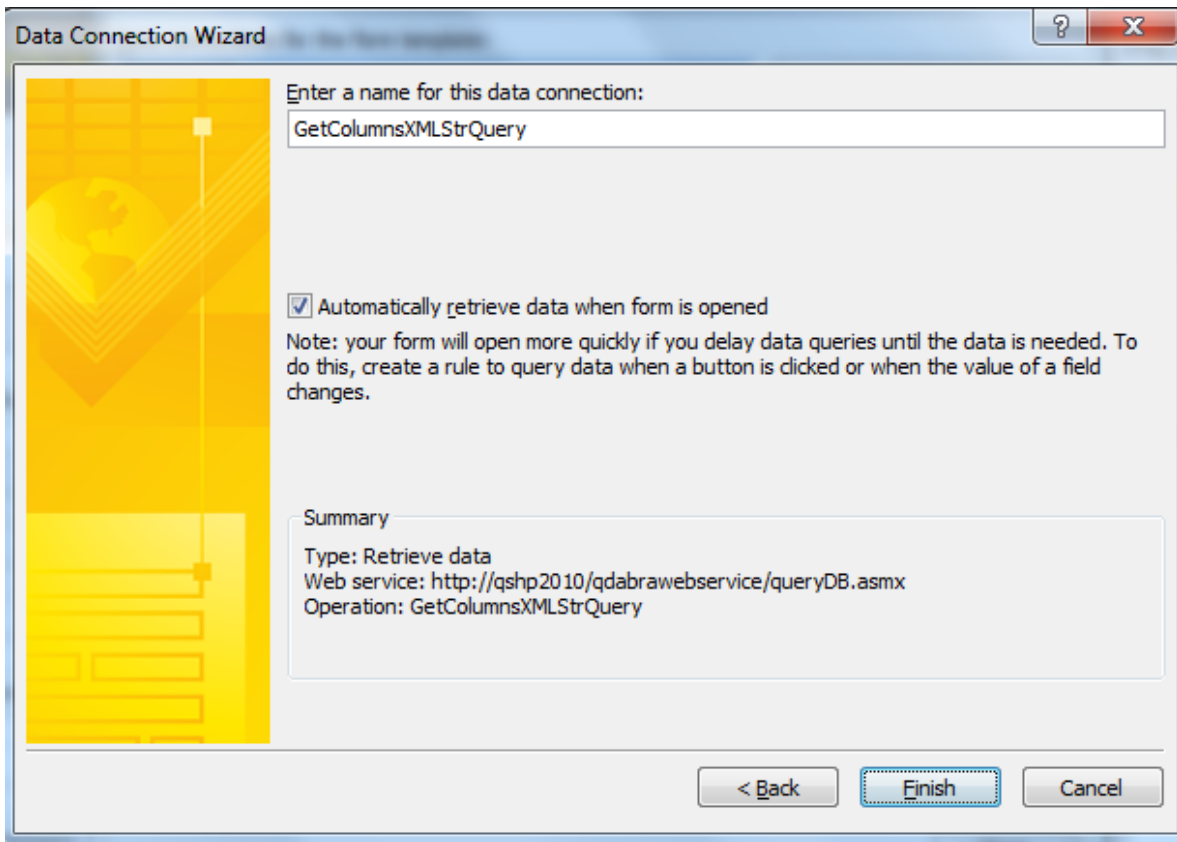
---

USE QUERYDB IN THE FORM

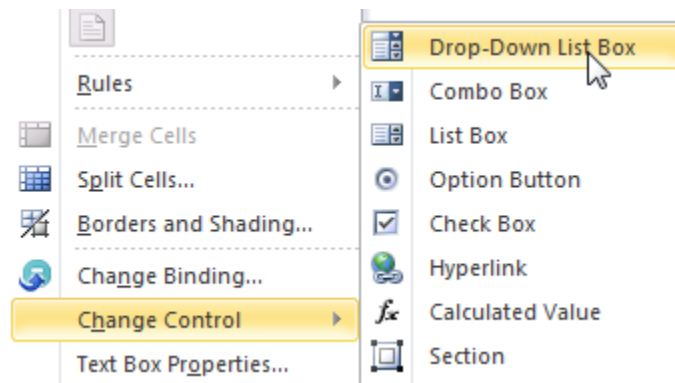
29. Back in the form in design mode, go to **Data > Data Connections**.
30. Click **Add**.
31. Select **Receive Data** and click **Next**.
32. Select **SOAP Web Service** and click **Next**.
33. Enter the URL for the QueryDB web service and click **Next**.



34. Select **GetColumnsXmlStrQuery** and click **Next**.
35. Click on Set Sample Value and paste the queryXML generated by the Query Builder tool.
36. Click **Next**, enter the same value into the next screen, and then click **Next** twice.
37. Click **Finish** and then click **Close**.



38. Right click on the Department field and select **Change Control > Drop Down List Box**.

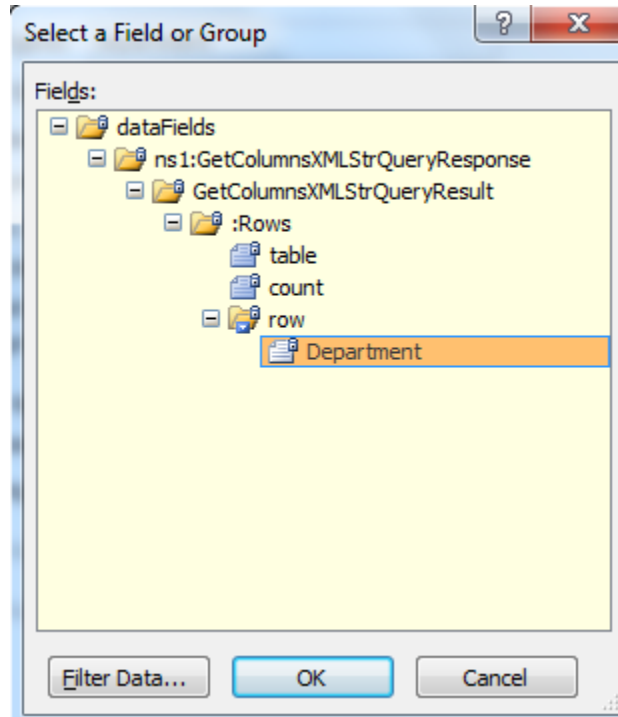


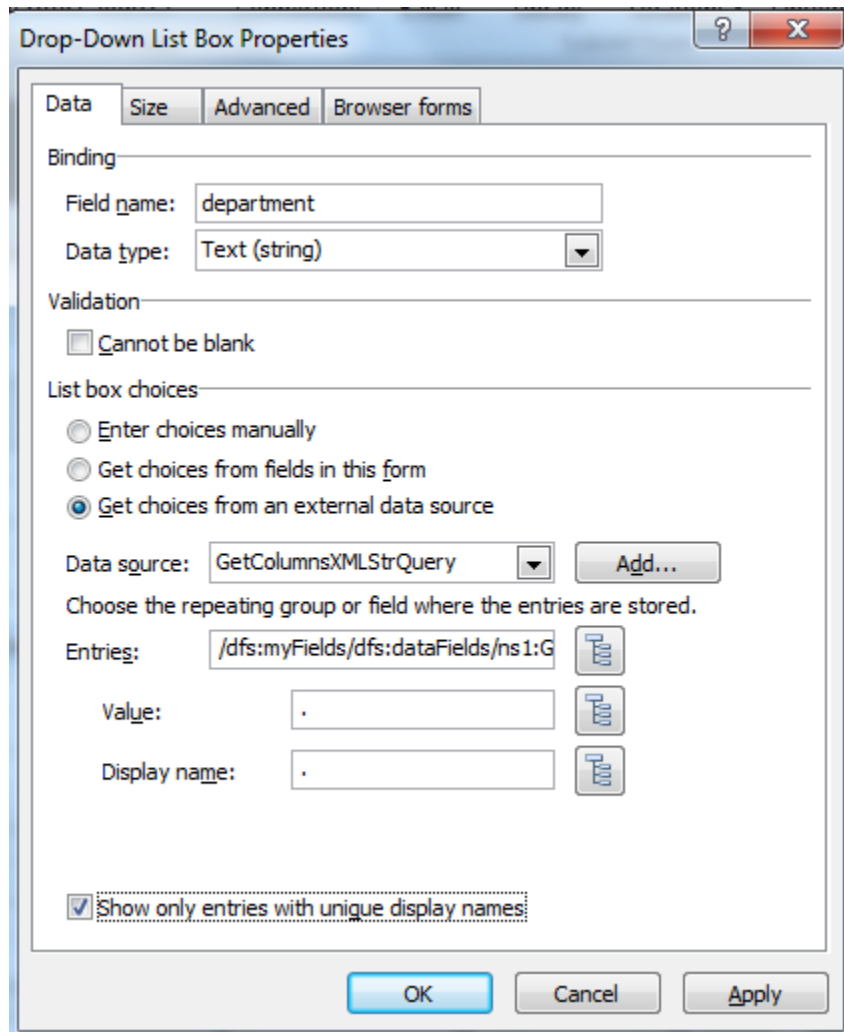
39. Right click on the dropdown and select **Properties**.

40. Under **List box choices**, find the **Data source** dropdown and select the new data connection to QueryDB.

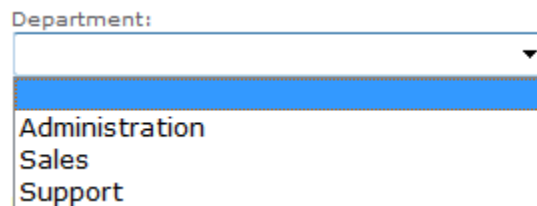
41. Select the **Department** node for **Entries**.

Qdabra DBXL Training: Module 5





42. Check the checkbox for “Show only entries with unique display names” and click **OK**.
43. Preview the form and verify that the dropdown shows the departments as expected:






---

### GENERATE A NEW QUERY

Return to the QueryBuilder form, in order to generate the next query.



## Qdabra DBXL Training: Module 5

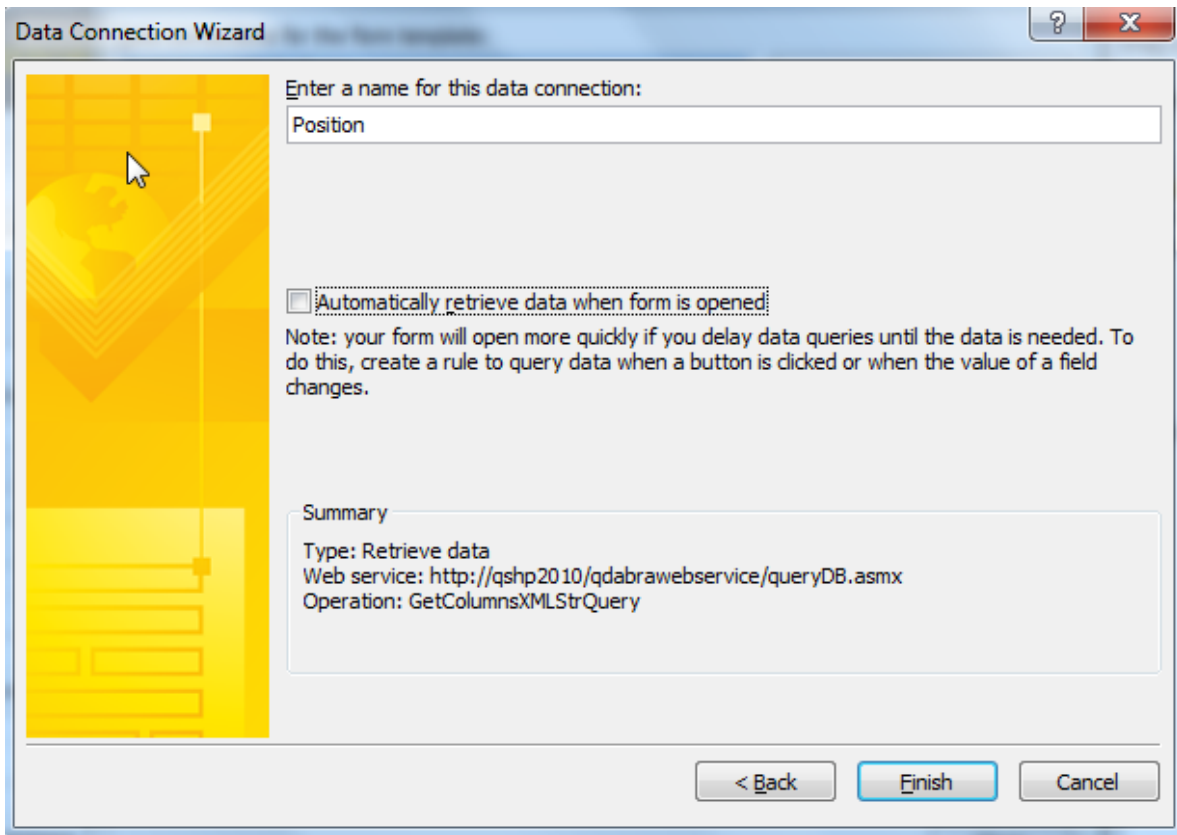
44. Under **Query Columns**, remove the row in that table.
45. Under **Data Sources**, attach the InfoPath form template (XSN) that we have been working with during training.
46. Under **Query Columns**, click on **Insert return column**.
47. Click the **Select Database Column** icon (  ).
48. In the taskpane, double click **Position**.
49. Under **Query Filter**, click on **Insert comparison**.
50. Click the **Select Database Column** icon.
51. In the taskpane, double click **Department**.
52. Under **Value or Column Name**, click the **Select Schema Node** icon (  ).
53. In the taskpane, double click the my:department node under my:employee.
54. Click **Build Now**. Copy the first and third queries into Notepad.
55. In Notepad, remember to change database='DBXLTraining' to database='#DBXLTraining#' to use the alias.

---

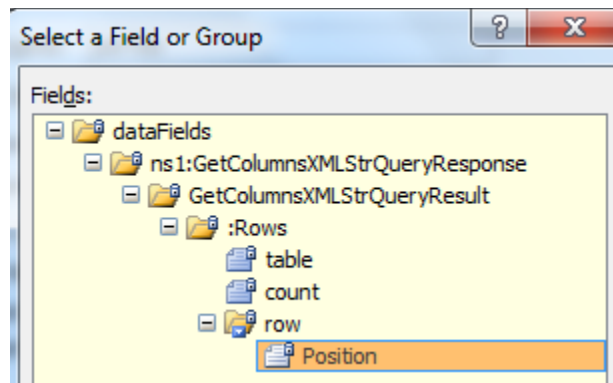
## MODIFY THE FORM

56. Back in the form in design mode, go to **Data > Data Connections**.
57. Click **Add**.
58. Select **Receive Data** and click **Next**.
59. Select **SOAP Web Service** and click **Next**.
60. Enter the URL for the QueryDB web service and click **Next**.
61. Select **GetColumnsXmlStrQuery** and click **Next**.
62. Click on Set Sample Value and paste the queryXML generated by the Query Builder tool.
63. Click **Next**, enter the same value into the next screen, and then click **Next** twice.
64. Click **Finish** and then click **Close**.

You'll notice that the data connection is named "GetColumnsXMLStrQuery 2" by default. This isn't very useful. In fact, the name of the first data connection was also not very helpful: GetColumnsXMLStrQuery. Ideally, name your data connections something useful and meaningful.

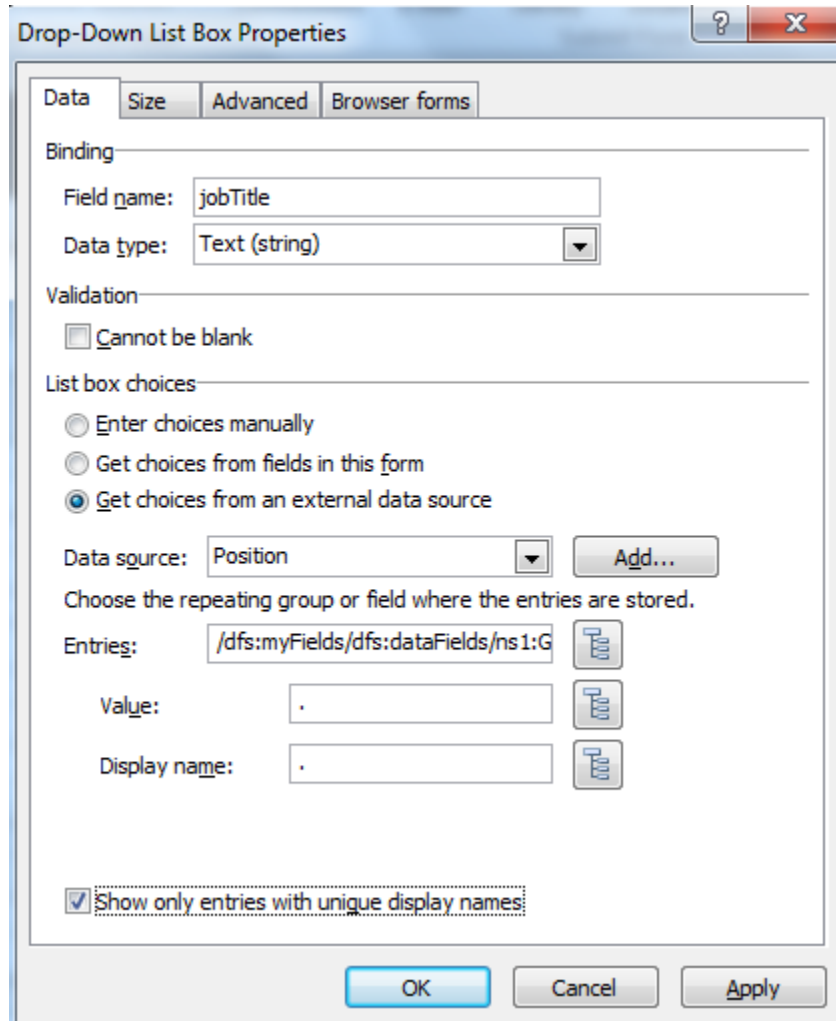


65. Right click on the Title text box and select **Change Control** > **Drop down list box**.
66. Right click on the Title text box and select **Properties**.
67. Under **List box choices**, select "Get choices from an external data source".
68. Select the second data connection created (which queries for Position) from the **Data source** dropdown.
69. For **Entries**, select the **Position** node.



Qdabra DBXL Training: Module 5

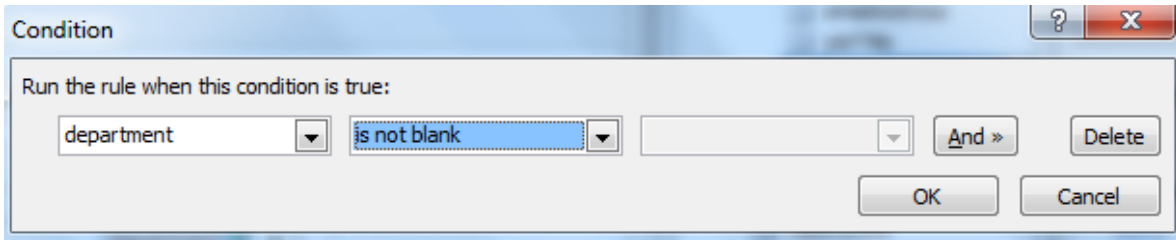
Your dropdown properties will look like this once you've completed these steps:



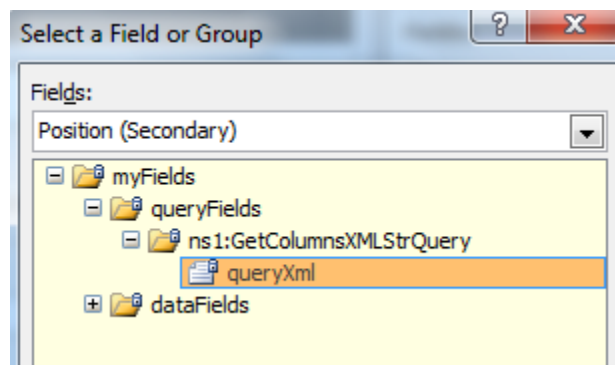
We have set everything up except the rules to query!

70. Select the **Department** dropdown and then select **Home > Manage Rules**.
71. Click **New > Action**.
72. Click the blue text under **Condition**.
73. The condition you will add is "department" in the first dropdown and "is not blank" in the second dropdown, as shown below.

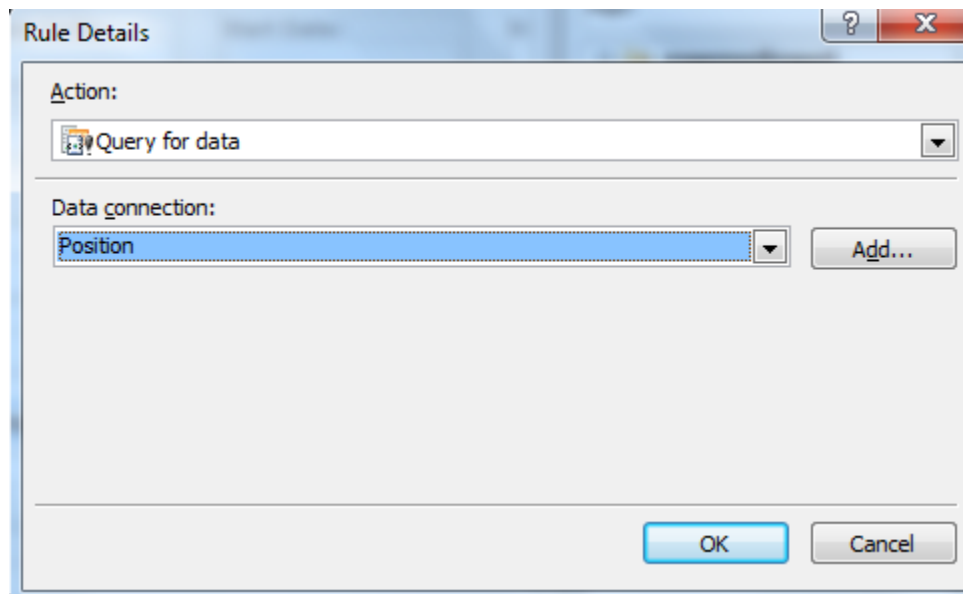
Qdabra DBXL Training: Module 5



- 74. Click **Add > Set a field's value**.
- 75. For the **Field**, select the queryXml node in the **Position** data connection.



- 76. For the **Value**, click the fx button and then paste the third query (which you pasted previously into Notepad).
- 77. Click **OK** twice.
- 78. Select **Add > Query for data** and select the Position data connection.



Qdabra DBXL Training: Module 5

Let's fill out the form to understand what we've accomplished.

79. Click **Preview**.

80. When the form loads, select a value from the Department dropdown. This will load options into the Position dropdown. We've implemented a simple case of cascading dropdowns using QueryDB.

<p>Department: <input type="text" value="Administration"/></p> <p>ID Number: <input type="text"/></p>	<p>Title: <input type="text"/></p> <p>Accountant HR</p>
<p>Department: <input type="text" value="Sales"/></p> <p>ID Number: <input type="text"/></p>	<p>Title: <input type="text"/></p> <p>Sales manager Sales representative</p>
<p>Department: <input type="text" value="Support"/></p> <p>ID Number: <input type="text"/></p>	<p>Title: <input type="text"/></p> <p>Support engineer Support manager</p>

If you had a large data set in SQL, this method is very useful in improving performance, because QueryDB filters on the server (DBXL web service) and not in the client (InfoPath form).