

Website Pros Database Component

v1.00.02

Table Of Contents

Before Getting Started.....	2
Using the Database Component.....	5
How the Database Component Works.....	5
Adding the Toolbar.....	6
Adding Component Objects.....	7
Setting Object Attributes.....	9
Using Editors.....	11
What Are Editors?.....	11
Using the Schema Editor.....	12
Using the SQL Wizard.....	15
Using the Expression Wizard.....	18
Operator Definitions.....	20
Component Object Definitions.....	21
Database Objects.....	21
Connector.....	21
Recordset.....	22
Insert.....	23
Delete.....	23
Update.....	23
Data Output Objects.....	24
Label.....	24
Image.....	24
Link.....	25
Textbox.....	25
TextArea.....	26
ComboBox.....	26
Hidden Field.....	26
Flow Control Objects.....	28
Conditional (If - Else - ElseIf).....	28
Recordset Iterator.....	28
Record Set Navigator.....	29
Tutorial.....	33
Setting Up the Database.....	34
Lesson 1. Data Retrieval.....	36
Building a Data Retrieval Application.....	36
Building a Query for the Home Page.....	38
Looping through a Recordset.....	40
Displaying Database Records on the Home Page.....	42
Navigating through a Recordset.....	44
Viewing the Contact List.....	46
Lesson 2. Inserting Data.....	47
Using a Form to Submit New Data.....	47
Inserting New Records.....	49
Viewing New Records.....	52

Table Of Contents

Lesson 3. Updating Data	53
Using a Form to Update Data.....	53
Using Variables and Where Clauses	55
Adding Conditional Expressions.....	58
Updating Records.....	62
Viewing Updated Records	65
Getting Help	67
Index.....	69



You can use the **Website Pros Database Component** to develop dynamic, full-featured applications on ColdFusion, ASP, and PHP platforms. Incorporate common dynamic elements in objects such as forms, lists, and tables, all from within the NetObjects Fusion user interface and build pages that will:

- List details of a particular record to display the name, price, and description of a particular product.
- List all records in a database table to display names or all available products.
- Link each record in a list either to details about that record, so that clicking a product name displays a price and description, or to a related list from another table, to display all stores where the product is available.

You can do all this by simply dragging and dropping components on the page, setting attributes in the object properties palette, and using wizards to add tables and construct queries. With the Database Component, you won't need to write the code - the component will generate it for you.

To get started, check out **How the Database Component Works** and **Adding the Component Toolbar**.

Use the **Tutorial** to build an application where employees at Mountain Jacques Sports Company can view contact information of employees at different locations online. The tutorial will guide you through:

- Adding objects and setting attributes.
- Retrieving and displaying records.
- Using Where clauses and conditional statements.
- Using forms to insert and update new records.

Before Getting Started

Please review the following information before working with the Database Component.

1. The Database Component requires v.7.5 Update #3 or higher. Visit **www.netobjects.com** to view the product downloads page.
2. The code generated by the Database Component is evaluated on the server you publish to. It is important that you verify your server supports the database language you will be using. For information on server requirements, check with your hosting provider and/or platform documentation.
3. Databases and data sources should be created before using the Database Component.

Note that while some database applications, such as Microsoft Access and SQL Server, allow spaces in table names, the Database Component will allow table names with no spaces. (Ex. Name your table FirstName rather than First Name.)

4. The ASP and ColdFusion components connect to any database using ODBC.
5. The PHP component connects directly to any MySQL database or to an MS Access database on a Windows platform using a datasource. Full support for MS Access and MySQL are provided and can be selected in the PHP Connector Properties palette.

In addition, Adodb v3.05 drivers are available in the PHP components directory. To use one of these additional drivers, locate the correct file at `..\Components\DBComponent\db\php\000.Connector`, and then type the file name in the Custom Engine field in the PHP Connector Properties palette.

For databases that you don't see a driver you may get them from the Adodb v3.05 for PHP distribution and install to the following location:

`...\Components\DBComponent\db\php\000.Connector`

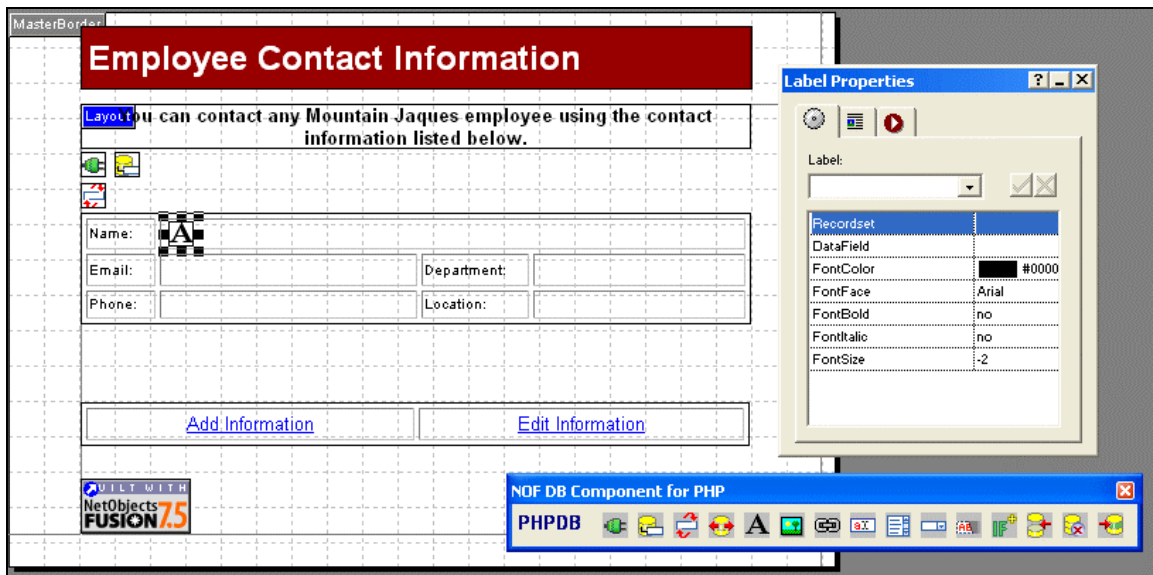
6. Verify your data source has been successfully established.

Website Pros offers a Tag Library Component that works with the Database Component to fully integrate dynamic elements into your Web site. To download the component, launch NetObjects Fusion and download the Tag Library Component from the Control Panel in your Online view.

Using the Database Component

How the Database Component Works

You build your ColdFusion, ASP, or PHP application by placing Database Component objects on one or more pages that display data from a database. Each page needs at least 2 objects - a **Connector** to identify the database the data will come from and an object to query the database. You can also add objects to display fields and allow site visitors to search for data, navigate forward and backward through results, and update, add, or delete records.

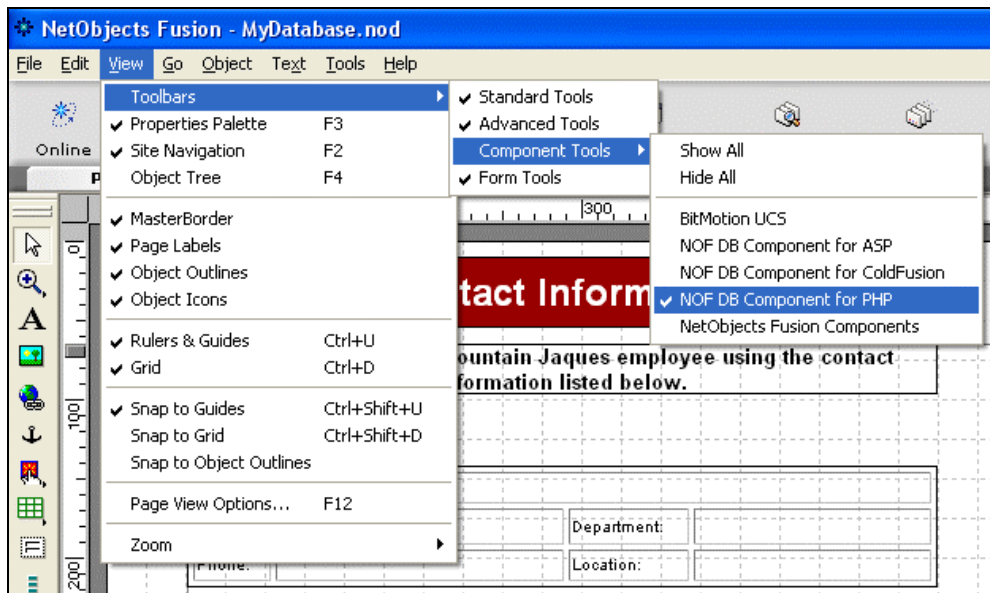


Once a component object has been added to your page, you set the attributes in the object properties palette. If you are unfamiliar with SQL and relational databases, you can use the **Editors** to add your tables and fields and build your queries.

Adding the Toolbar

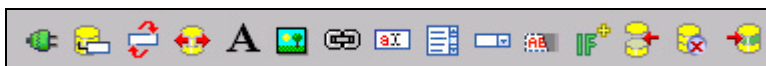
To add the Database Component toolbar:

1. In Page view, from the View menu, choose Toolbars>Component Tools.



2. Select the Database language you will be working with.

The Database Component Toolbar will appear.



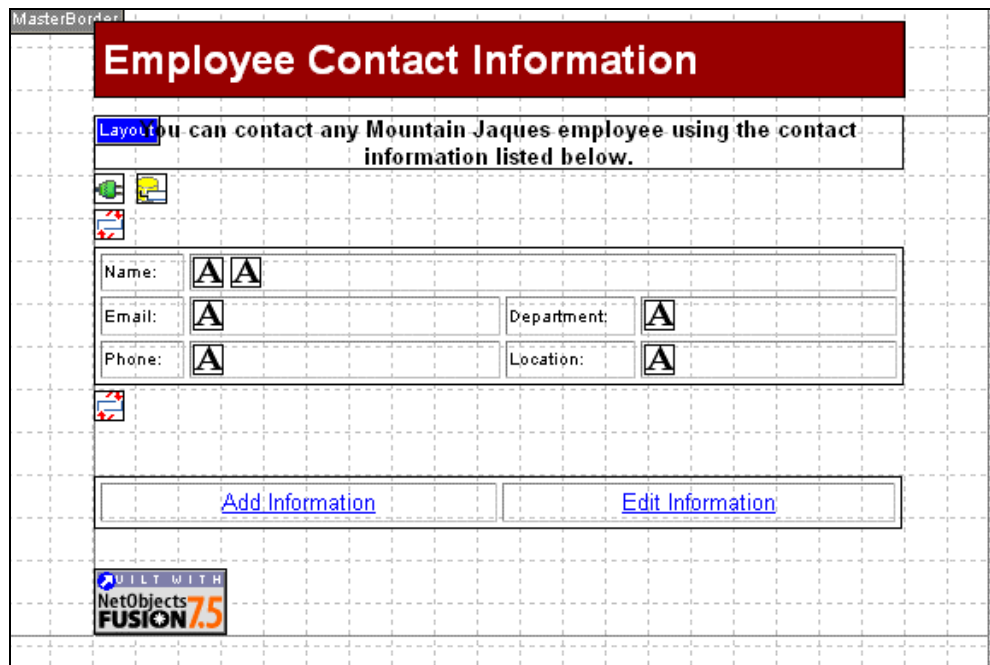
- To dock the toolbar, double-click its title bar or drag it to the left side or top of the window. You can move the toolbar by positioning the pointer over the double line at its top and dragging it anywhere in the window.
- To hide and show toolbars, from the View menu choose Toolbars > Components Tools > DB Language Name. A check mark on the menu indicates a Database Component toolbar is displayed.

Adding Component Objects

You can easily add component objects to any page of your site.

To add a component object:

1. In Page view, select an object from the Database Component Toolbar.
2. Click on the page where you want to place the object.



An object placeholder and the object Properties palette appear.

3. Specify the object attributes in the Properties Palette. See **Setting Object Attributes**.

When adding Database Component objects, remember that:

- A connector object must be placed on each page you will be using the component.

Only one connector can be added to a page.

- Objects must be added/placed in the correct order for the code to be generated correctly. For example, when using a Recordset Iterator object, make sure the "Start" iterator object is placed before the "end" iterator object. Otherwise, the site

Website Pros Database Component

will publish with errors and the database will not be functional.

It is a good idea to add objects in a table to make sure placement is correct and the records will be aligned correctly when displayed.

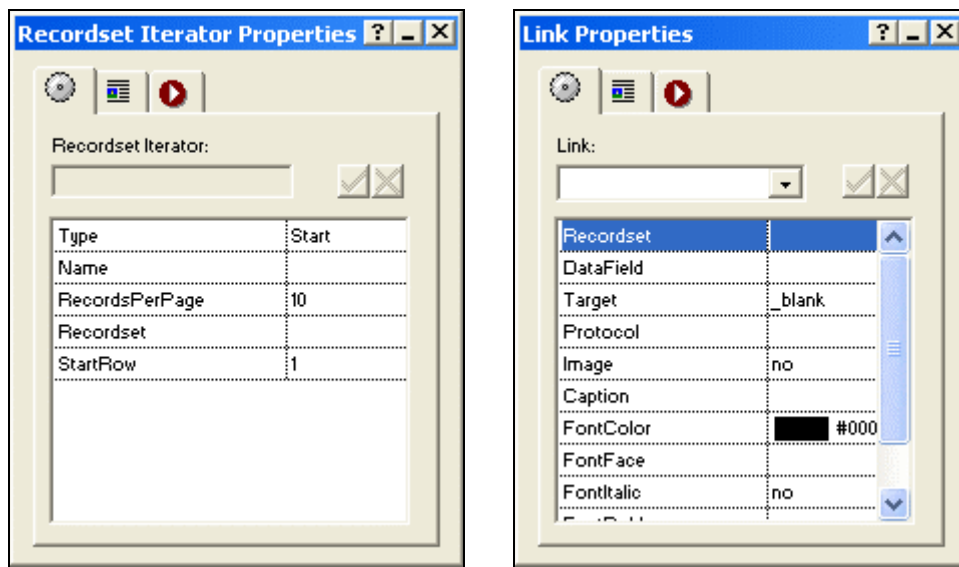
Setting Object Attributes

Object attributes are edited in the Properties palette. For specific attribute details, see the **Component Object Definitions**.

To set object attribute properties:

1. In Page view, add an object to the page. The object placeholder and the object Properties palette appear.

The attributes listed for each object will be different according to the object.



2. Set the object Type, if available. This attribute is used for objects needing a start/close tag or if there are different variations of the object, such as If-Else-ElseIf-EndIf. If the object does not need to be closed, the Type attribute will not appear.
3. Some attributes must be set before you can proceed.
 - If the object you are working with requires a name, type a unique name in the Name field.
 - If the object you are using requires that you add or define tables, fields and/or conditions, you can use an Editor. See **Using Editors**.
 - If the object you are working with requires that a recordset be defined, choose from the recordset field drop-down in the properties palette. After

Website Pros Database Component

the recordset is defined, you can select a data field from the data field field drop-down in the properties palette.

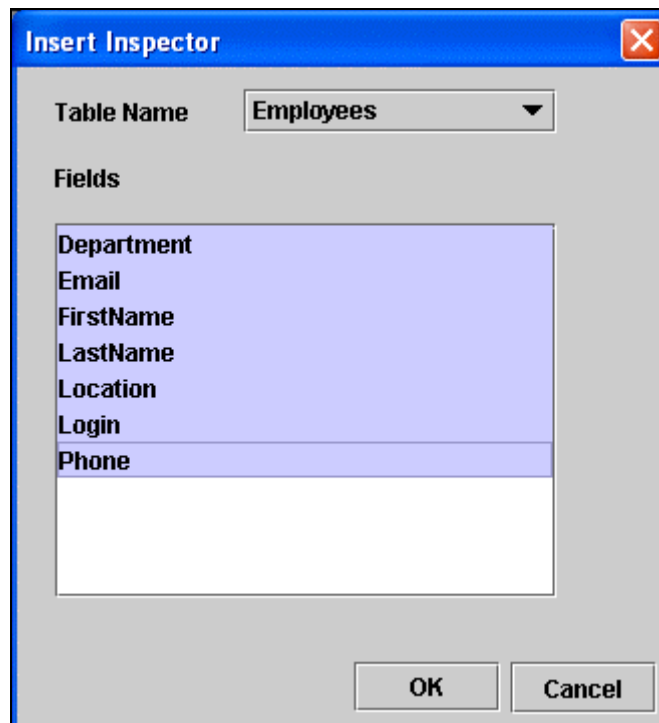
- If the object you are working with requires that a recordset iterator be defined, choose from the iteratorName field drop-down in the properties palette.
4. Complete the process by selecting the remaining attributes (ex, FontSize), enter the attribute's value in the Properties Palette and click the check mark.

NetObjects Fusion does not verify object attribute values, so be sure to use valid syntax. Many objects have attributes that require values for them to function correctly.

Using Editors

What Are Editors?

The Database Component also allows you to insert custom statements by typing directly in the object properties palette. However, if you are unfamiliar with SQL and relational databases, you can use editors to easily add tables, define variables, and filter queries.



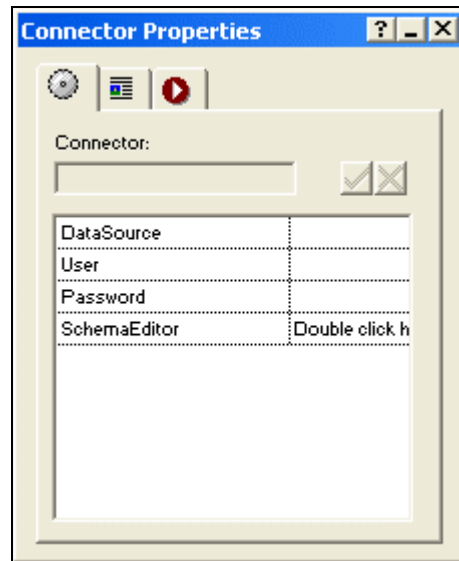
There are 3 editors available in the Database Component:

- **Schema Editor** to assist in setting up your tables, fields, and variables.
- **SQL Wizard** to assist in adding tables and specifying records from which to search from.
- **Expression Wizard** to assist in building conditional expressions to filter recordsets.

Using the Schema Editor

The Schema Editor is used with the Connector object.

The **Schema Editor** is used to configure your tables and variables. These values only need to be configured in the Schema Editor and will be stored so they can be selected in the SQL and Expression Wizards.



The tables you add on the **DB Schema** tab will be used to build a recordset. After adding a table, you will need to determine which data fields from that table you want to pull records from to form your recordset. The order in which you add your tables and fields is not important in the Schema Editor - by default, the sorting will be done in alphabetical, ascending order unless otherwise specified in the Recordset Properties Palette. There is no limit to the number of tables and fields you can add in the Schema Editor.

On the **Variables** tab, you can define the following variable types:

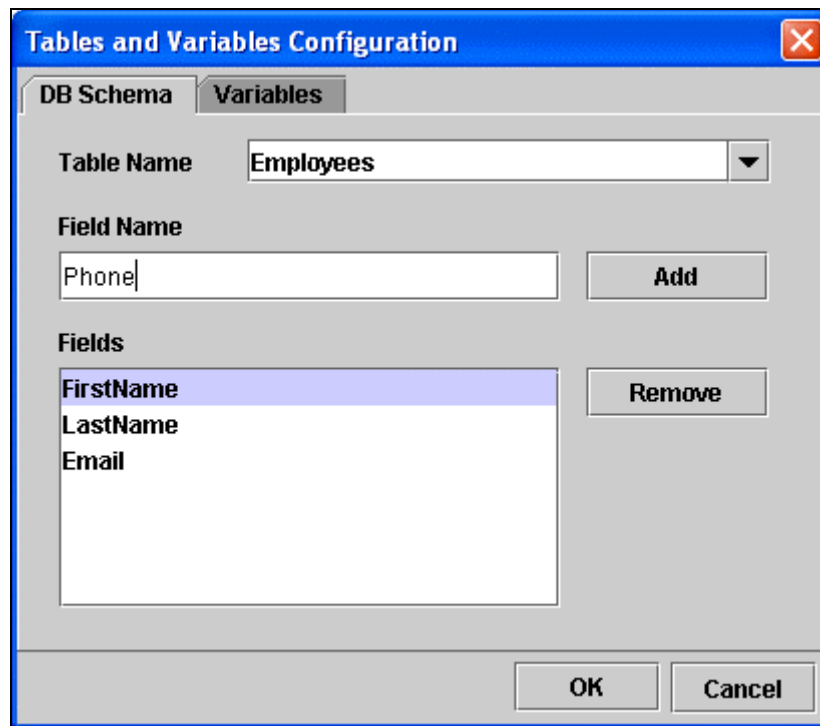
- **URL Parameter:** Variables reflecting parameters appended to a URL.
- **CGI Environment:** Variables reflecting the CGI environment variables of the context of the requested template.
- **Form:** Variables reflecting the values of the fields in a form submitted to the current template.
- **Client:** Variables used to associate data with a specific client.
- **Session:** Variables used to associate data with a given client session.
- **Server:** Variables used to associate data with a given client session.
- **Application:** Variables used to associate data with a specific application.
- **Cookie:** Variables reflecting cookies passed by the requesting browser.

Note that you must first add a Data Source in the properties palette before you can configure the tables and variables.

Configuring Tables

To configure tables using the Schema Editor:

1. Click on the DB Schema tab.
2. In the Table Name field, enter the name of the table you will be using as it appears in your database.
3. In the Field Name field, enter the name of a field that will be queried as it appears in your table.
4. Click Add. The field name will appear in the Fields list and will be associated with the table selected in the Table Name drop-down list.



5. Repeat Steps 2 and 3 to add additional data fields that will be queried.
6. To add additional tables, type a new name in the Table Name field and add data fields.

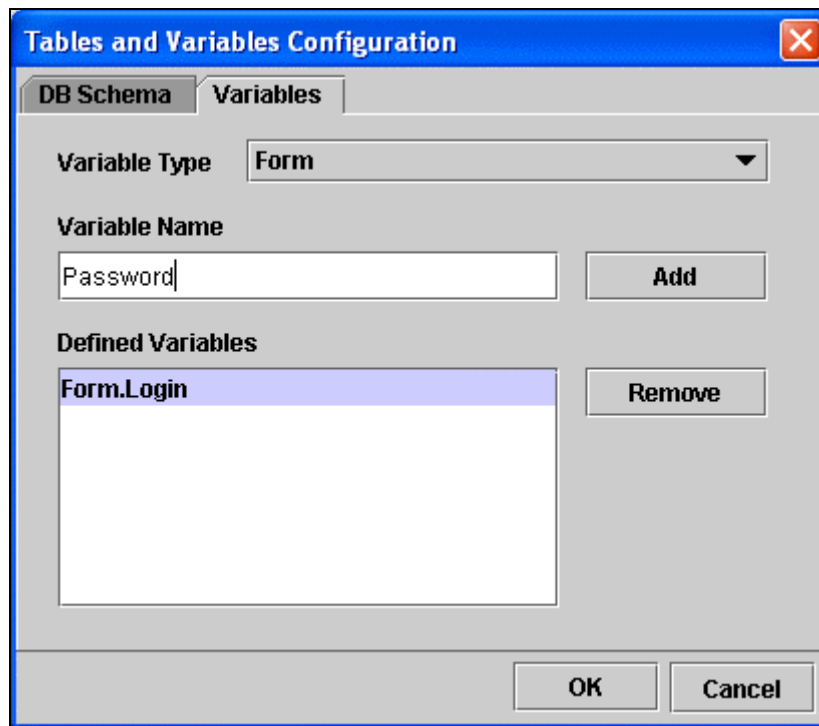
Note You must add a field in order for the table to be stored

7. Click OK to close the Schema Editor.

Configuring Variables

To configure variables using the Schema Editor:

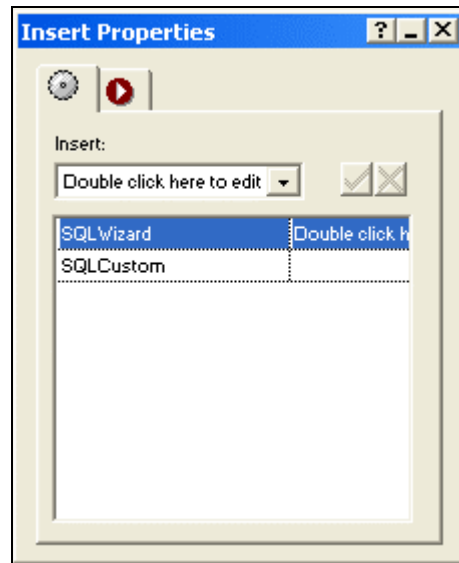
1. Click on the Variables tab.
2. Select a variable type from the drop-down list:



3. In the Variable Name field, enter a name for the new variable.
4. Click Add. The variable will appear in the Defined Variables list.
5. Repeat Steps 2 and 3 to add additional variables used in queries.
6. Click OK to close the Schema Editor.

Using the SQL Wizard

The SQL Wizard is used with the Recordset, Insert, Delete and Update objects.



Use the **Fields** tab in the SQL Wizard to select specific datafields to query. If you added your tables and datafields in the Schema Editor, the datafields will appear after choosing a table. For instance, if you have a table named "Students" and you want to query the records by last name, major, and classification, you would highlight the "LastName", "Major", and "Classification" datafields listed.

You can also use the **Where Clause** tab to narrow your search results. You use this when you want to return records that match a particular set of criteria. So, rather than pull records for all students in your class, you can narrow your results by searching for "Biology" in the Major datafield, which in turn, will return only those records meeting that criteria.

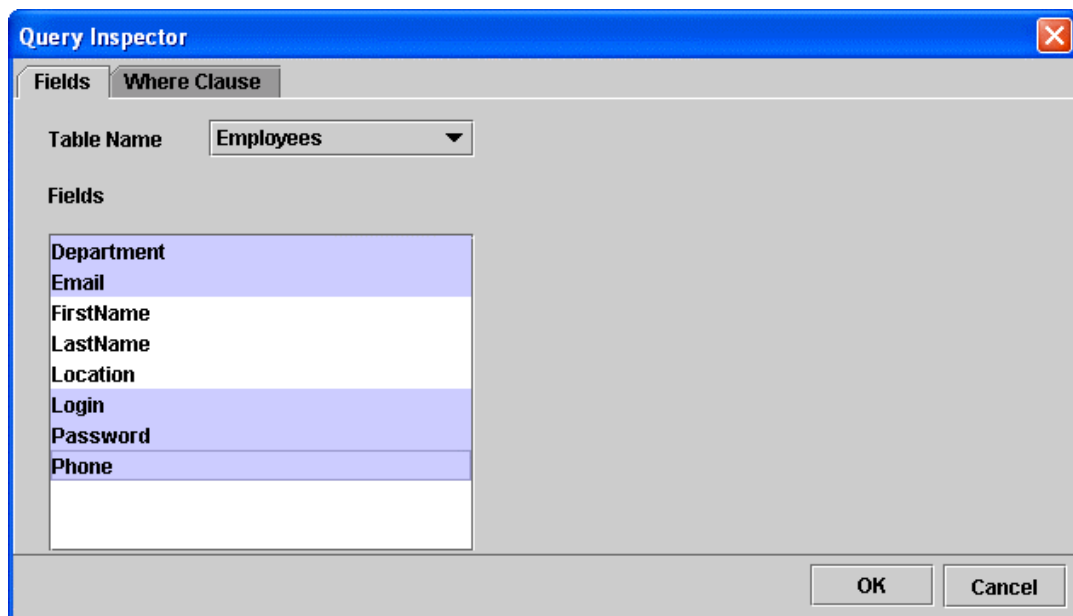
The where clause is not required; however, using a where clause is recommended. If you have, for example, 10,000 rows of data, it is likely you won't want to return them all. You can use the Where clause to filter your results to a manageable number. In addition, you can use the where clause to view the impact of updating or deleting records before actually carrying out the action. For example, if you are going to delete records, first use the Where clause to display the records you will be deleting. If you see that the data you will be deleting is wrong, you can make adjustments before taking any actions that cannot be undone.

Adding Tables and Fields

To add tables and data fields using the Query Inspector:

1. Click on the Fields tab.
 - If using the **Insert** object, go directly to Step 2.
 - If using the **Delete** object, go directly to **Filtering Results**.
2. Select a table from the "Table Name" drop-down list.
3. Highlight the datafields you want to use in the Fields list.

To select more than one field, press CTRL and select multiple fields.



YOU MUST HIGHLIGHT DATA FIELDS BEFORE CLOSING THE EDITOR OR THEY WILL NOT APPEAR IN OBJECT PROPERTIES PALETTES.

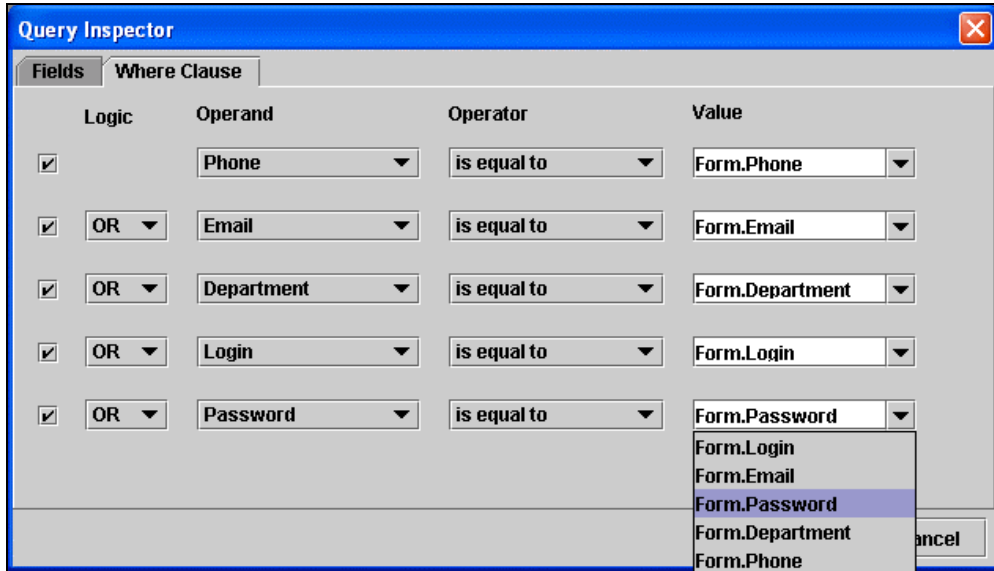
4. Click OK to close the SQL Wizard.

Filtering Results

To narrow the query result set using the Query Inspector:

1. Click on the Where Clause tab.
 - If using the **Delete** object, go to Step 2

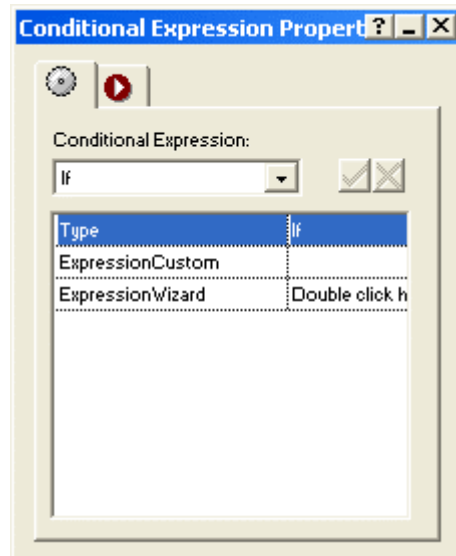
- If using the **Insert** object, go to "Add Tables and data fields"
2. Click in the check box to activate the clause.
 3. In the Operand field, select the table
 4. In the operator field, select an operator from the drop-down list.



5. In the Value field, either type in a value or choose a pre-defined value from the drop-down list.
6. To narrow the results further, click on the next checkbox and repeat Steps 3 through 5.
7. Click OK to close the SQL Wizard.

Using the Expression Wizard

The Expression Wizard is used with the If and ElseIf objects.



The **Expressions Wizard** is used to build conditional statements to further narrow your query results. The Where clause filters records at the database level; the conditional statement queries your recordset results. For example, you have queried your "Students" table and returned 20 records of students majoring in Biology. Now, you would like to display these records on your Web site - however, you want to display senior records using a red font to set them apart. In this case, you can build a conditional expression by searching the Classification datafield for "seniors". The resulting expression would be:

```
If Classification = "senior"  
    font = red  
Else  
    font = black  
EndIf
```

You can add multiple conditions to one expression by including an ElseIf object. If you wanted to display freshman records in blue font, the resulting expression would be:

```
If Classification = "senior"  
    font = red  
ElseIf Classification = "freshman"  
    font = blue  
Else  
    font = black  
EndIf
```

Notice that each expression has a beginning object (If) and a closing object (EndIf).

Adding Conditions

To construct conditional statements using the Conditional Expressions Editor:

1. Click in the check box to activate the expression.
2. In the Operand field, select the variable.

Logic	Operand	Operator	Value
<input checked="" type="checkbox"/>	updates.Login	is equal to	Form.Login
<input checked="" type="checkbox"/>	updates.Password	is equal to	Form.Password
<input type="checkbox"/>	Form.Login	is empty	Form.Login
<input type="checkbox"/>	Form.Login	is empty	updates.Phone
<input type="checkbox"/>	Form.Login	is empty	updates.Department
<input type="checkbox"/>	Form.Login	is empty	updates.Login
<input type="checkbox"/>	Form.Login	is empty	updates.Email
<input type="checkbox"/>	Form.Login	is empty	Form.Password
<input type="checkbox"/>	Form.Login	is empty	updates.Password

3. In the operator field, select an operator from the drop-down list.
4. In the value field, you can:
 - Select a variable from the drop-down list.
 - Type in a new value.
5. To narrow the results further, click in checkbox on the next line and repeat Steps 3 through 5.
6. Click OK to close the Conditional Expressions Editor.


Operator Definitions

Operator	Description
and	All conditions must be met or no results are returned.
or	At least one condition must be met or no results are returned.
is empty	Tests that the value on the left side is empty.
is not empty	Tests that the value on the left side contains data.
is equal to	Tests that values match completely or no results are returned.
is not equal to	Tests that values are different.
is lower than	Tests that the value on the left side of the operator is less than the value on the right side.
is lower or equal	Tests that the value on the left side of the operator is less than or equal to the value on the right side.
is greater than	Tests that the value on the left side of the operator is greater than the value on the right side.
is greater or equal	Tests that the value on the left side of the operator is greater than or equal to the value on the right side.
contains	Tests that the value on the left side of the operator includes the value on the right side.
starts with	Tests that the value on the left side of the operator starts with the value on the right side.
ends with	Tests that the value on the left side of the operator ends with the value on the right side.
does not contain	Tests that the value on the left side of the operator does not include the value on the right side.

Component Object Definitions

Database Objects

Connector

 The CONNECTOR object must be used on each page in which a component object is used. The connector "points" to the physical database you created, allowing you to access the data and query specific tables and fields. Note that the Connector attributes differ according to the platform being used.

If using ASP and ColdFusion...

Attribute	Description	Required?
DATASOURCE	The data source name (DSN) created to connect to the database containing your table(s). The data source must be created before connecting to the physical database.	Required
USER	Username required to access your database.	Optional
PASSWORD	Password corresponding with the username required to connect to your database.	Optional
SCHEMA EDITOR	Use an SQL Editor to configure tables and variables. See Using the Schema Editor.	Optional


If using PHP...

Attribute	Description	Required?
DATASOURCE	The data source name (DSN) created to connect to the database containing your table(s). The data source must be created before connecting to the physical database. If using a direct connection, the datasource will be used to associate table and field data with this profile.	Required
DB ENGINE	The engine used to create your database. Options are: <ul style="list-style-type: none"> ▪ MS Access ▪ MySQL 	Required

Website Pros Database Component


CUSTOM ENGINE	Custom engine, such as Oracle or Sybase, used to create your database. See Before Getting Started.	Optional
DATABASE	The name of the Database being queried.	Required
HOST	The name of the computer on which your program is running.	Required
USER	Username required to access your database.	Optional
PASSWORD	Password corresponding with the username required to connect to your database.	Optional
CONNECTION TYPE	The type of connection used to connect to your database. Options are: <ul style="list-style-type: none">▪ Direct▪ By DataSource	Required
SCHEMA EDITOR	Use an SQL Editor to configure tables and variables. See Using the Schema Editor.	Optional

Recordset

 The RECORDSET object is used to define the set of results a query will return. The recordset is used by Data Output objects to access and display data.


Attribute	Description	Required?
NAME	Name for the query result recordset.	Required
SQL WIZARD	Use an SQL Editor to add tables and data fields, and filter the query result set. See Using the SQL Wizard.	
ORDER BY	Orders query results by the first specified data field.	Optional
SORT	Sorts query column data. Default is Ascending.	Optional
SQL CUSTOM	Add SQL statements to define tables, data fields and filter the query result set.	Optional
MAXROWS	The maximum number of rows returned in the recordset. If no number is specified, all rows will be returned. Default is -1.	Optional

Insert

 The INSERT object is used to insert new records in a table.


Attribute	Description	Required?
SQL CUSTOM	Add SQL statements to define tables and data fields.	Optional
SQL WIZARD	Use an SQL Editor to add tables. See Using the SQL Wizard	

Delete

 The DELETE object is used to delete records from a table.

Attribute	Description	Required?
SQL CUSTOM	Add SQL statements to define the table and filter the query result set.	Optional
SQL WIZARD	Use an SQL Editor to define the table and filter the query result set. See Using the SQL Wizard	


Update

 The UPDATE object is used to update existing records in tables after modifications have been made.

Attribute	Description	Required?
SQL CUSTOM	Add SQL statements to define tables, data fields, and filter the query result set.	Optional
SQL WIZARD	Use an SQL Editor to add tables and filter the query result set. See Using the SQL Wizard	

Data Output Objects

Label

 The LABEL object is used to display results of a query.

Attribute	Description	Required?
DATA FIELD	The database field in which the query will search.	Required
FONT BOLD	Displays the query results in bold.	Optional
FONT COLOR	Displays the query results in the designated text color.	Optional
FONT FACE	Displays the query results in the designated style.	Optional
FONT ITALIC	Displays the query results in italics.	Optional
FONT SIZE	The font size, in pixels.	Optional
RECORDSET	The recordset being queried.	Required

Image

 The IMAGE object displays images datafields.


Attribute	Description	Required?
ALT	The alt tag for the image.	Optional
BORDER	The image border in pixels.	Optional
DATA FIELD	The database field in which the query will search.	Required
HEIGHT	The image height in pixels.	Optional
WIDTH	The image width in pixels.	Optional
RECORDSET	The recordset being queried.	Required

Link

 The LINK object displays hyperlink data fields.

Attribute	Description	Required?
DATA FIELD	The database field in which the query will search.	Required
RECORDSET	The recordset being queried.	Required
IMAGE	The link to the image.	Optional
IMAGE SRC	Path to the image link.	Optional
CAPTION	The caption linking to the text.	Optional
FONT COLOR	Displays the caption in the designated color.	Optional
FONT FACE	Displays the caption in the selected style.	Optional
FONT ITALIC	Displays the caption in italics.	Optional
FONT BOLD	Displays the caption in bold.	Optional
FONT SIZE	Displays the caption in the selected size.	Optional

Textbox

 The TEXTBOX object is a form element providing a textbox field based on a query.


Attribute	Description	Required?
DATA FIELD	The database field in which the query will search.	Required
MAX CHARS	The maximum number of characters displayed.	Optional
NAME	The name of the text box.	Required
RECORDSET	The recordset being queried.	Required

TextArea

 The TEXTAREA object is a form element providing a text area based on a query.


Attribute	Description	Required?
DATA FIELD	The database field in which the query will search.	Required
NAME	The name of the Combo box.	Required
RECORDSET	The recordset being queried.	Required

ComboBox

 The COMBOBOX object is a form element that displays a recordset in a drop-down or list box.

Attribute	Description	Required?
NAME	The name of the ComboBox element.	Required
RECORDSET	The recordset being queried.	Required
DATA FIELD	The database field in which the query will search.	Required
LINES	The number of records displayed at one time in a ListBox. Default is 1.	Optional
CONTROL STYLE	Displays the recordset as a drop-down list or list box.	Optional

Hidden Field

 The HIDDENFIELD object is a hidden object placed on the page which the site visitor cannot see.


Attribute	Description	Required?
-----------	-------------	-----------

Component Object Definitions

DATA FIELD	The database field in which the query will search.	Required
NAME	The name of the Hidden Field element.	Required
<u>RECORDSET</u>	<u>The recordset being queried.</u>	<u>Required</u>


Flow Control Objects

Conditional (If - Else - Elseif)

 The CONDITIONAL object is used to construct simple and compound conditional statements. IF conditions are used with ELSE and ELSEIF conditions. Note that an If statement must be closed with an EndIf tag or an error will be generated after publishing.

Attribute	Description	Required?
TYPE	Select tag type: <ul style="list-style-type: none"> ▪ If, Else, or ElseIf to start the conditional statement. ▪ EndIf to close the conditional statement 	Required
EXPRESSION CUSTOM	Add SQL statements to construct conditional statements.	Optional
EXPRESSION WIZARD	Use an SQL Editor to construct conditional statements. See Using the Expression Wizard.	


Recordset Iterator

 The RECORDSET ITERATOR object is used to loop through a query record set. During each iteration, the columns of the current row are available for output. The iterations will continue until one or more conditions are met.

Attribute	Description	Required?
TYPE	Select: <ul style="list-style-type: none"> ▪ Start to begin the iterator tag. ▪ End to close the iterator tag. 	Required
NAME	Name for the iterator.	Required
RECORDS PER PAGE	The number of records displayed after the iteration(s) is complete. If no number is specified, all records will be displayed. Default is 10.	Optional.
RECORDSET	The recordset being queried.	Required
START ROW	The first row of the query that is included in the loop.	Optional

Default is 1.

Record Set Navigator

 The RECORDSET NAVIGATOR object is used to place smart links on pages displaying query results. The smart links allow site visitors to navigate forwards and backwards through the query results.

Attribute	Description	Required?
Iterator Name	The iterator being used.	Required
Alignment	Aligns the smart links on the page	Optional
Spacing	Space, in pixels, displayed between the smart links	Optional
FirstVisible	Places a smart link on the page taking the user to the first record of the query set result.	Optional
FirstImage	Displays the smart link to the first record if the query set result as an image.	Optional
FirstImageSrc*	Location of the image or file displayed as the FirstImage smart link.	Optional
FirstCaption	Caption used for the FirstImage smart link	Optional
FirstFontColor	Font color used for the FirstCaption text.	Optional
FirstFontFace	Font type used for the FirstCaption text.	Optional
FirstFontItalic	Italicizes the FirstCaption text.	Optional
FirstFontBold	Bolds the FirstCaption text.	Optional
FirstFontSize	Font size of the FirstCaption text.	Optional
PreviousVisible	Places a smart link on the page taking the user to the previous record of the query set result.	Optional
PreviousImage	Displays the smart link to the previous record if the query set result as an image.	Optional
PreviousCaption	Caption used for the PreviousImage smart link	Optional
PreviousFontColor	Font color used for the PreviousCaption text.	Optional

Website Pros Database Component

PreviousFontFace	Font type used for the PreviousCaption text.	Optional
PreviousFontItalic	Italicizes the PreviousCaption text.	Optional
PreviousFontBold	Bolds the PreviousCaption text.	Optional
PreviousFontSize	Font size of the PreviousCaption text.	Optional
PreviousImageSrc*	Location of the image or file displayed as the PreviousImage smart link.	Optional
NextVisible	Places a smart link on the page taking the user to the next record of the query set result.	Optional
NextImage	Displays the smart link to the next record if the query set result as an image.	Optional
NextCaption	Caption used for the NextImage smart link	Optional
NextFontColor	Font color used for the NextCaption text.	Optional
NextFontFace	Font type used for the NextCaption text.	Optional
NextFontItalic	Italicizes the NextCaption text.	Optional
NextFontBold	Bolds the NextCaption text.	Optional
NextFontSize	Font size of the NextCaption text.	Optional
NextImageSrc*	Location of the image or file displayed as the NextImage smart link.	Optional
LastVisible	Places a smart link on the page taking the user to the last record of the query set result.	Optional
LastImage	Displays the smart link to the last record if the query set result as an image.	Optional
LastCaption	Caption used for the LastImage smart link	Optional
LastFontColor	Font color used for the LastCaption text.	Optional
LastFontFace	Font type used for the LastCaption text.	Optional
LastFontItalic	Italicizes the LastCaption text.	Optional
LastFontBold	Bolds the LastCaption text.	Optional
LastFontSize	Font size of the NextCaption text.	Optional

LastImageSrc*	Location of the image or file displayed as the LastImage smart link.	Optional
---------------	--	----------

**Displayed as an attribute only if an image is being used for the smart link.*

Tutorial

Your employer has requested that you create a Web site displaying contact information for employees at all of the company locations. So, you are going to create an application consisting of pages that display data directly from the database where site visitors can see employees contact information, as well as add and modify employee records.

Employee Contact Information			
You can contact any Mountain Jaques employee using the contact information listed below.			
Name:	Jane Bennett		
Email:	jbennett@mjsports.com	Department:	Human Resources
Phone:	407-555-5678	Location:	Orlando
Name:	Ted Davis		
Email:	ted@@mjsports.com	Department:	Customer Service
Phone:	305-777-5678	Location:	Miami

To complete the lesson, you should already be familiar with the basic NetObjects Fusion concepts including creating sites from templates.

Setting Up the Database

Before building your database application, you will need to first create your database and then create a data source.

1. Create the database and name it *Tutorial*.
2. You will create a table that stores employee contact information.

Name the table *Employees* and use the following data.

Location	FirstName	LastName	Department	Email	Phone	Login	Password
Jacksonville	John	Smith	IT	jsmith@mjsports.com	904-444-1234	jsmith	jsmith
Orlando	Bill	Jones	Marketing	bjones@mjsports.com	407-555-1234	bjones	bjones
Miami	Mary	White	IT	mary@mjsports.com	305-777-1234	mwhite	mwhite
Jacksonville	Paula	Moore	Customer Service	pmoore@mjsports.com	904-444-5678	pmoore	pmoore
Tampa	Carl	Williams	Finance	cwilliams@mjsports.com	831-999-1234	cwilliams	cwilliams
Tampa	Lisa	Jackson	Marketing	ljackson@mjsports.com	831-999-5678	ljackson	ljackson
Orlando	Jane	Bennett	Human Resources	jbennett@mjsports.com	407-555-5678	jbennett	jbennett
Miami	Ted	Davis	Customer Service	ted@mjsports.com	305-777-5678	tdavis	tdavis

Note: MS Access users can download the *Employees* table [here](#). If using a printed manual, please go to the Online help in your NetObjects Fusion Control Panel and download from the Tutorial section.

4. Create a Data Source and name it *tutorial*.
5. Click [Download](#) to save the tutorial template.

Note: If using a printed manual, please go to the Online help in your NetObjects Fusion Control Panel and download from the Tutorial section.

6. Add the database component toolbar that corresponds with the platform you will be publishing to.

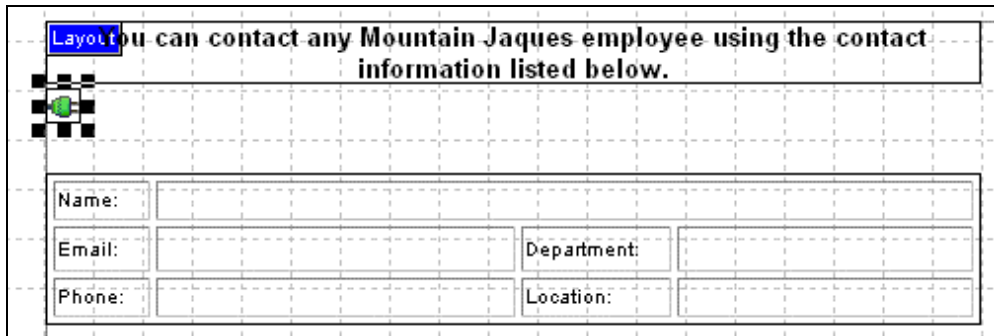
You are ready to start building your Web site application!

Lesson 1. Data Retrieval

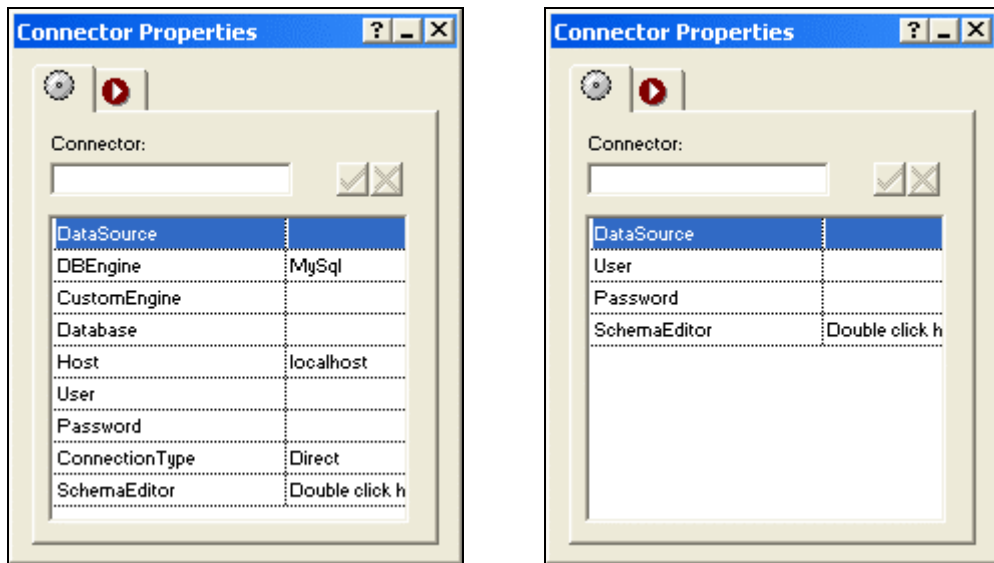
Building a Data Retrieval Application

In this lesson, you'll use an editor to add your tables and fields, and connect them to the *Tutorial* database created in the introduction.

1. From the toolbar, place a Connector object on the Home page. This will establish a connection to *Tutorial*.



The Connector Properties palette will appear.

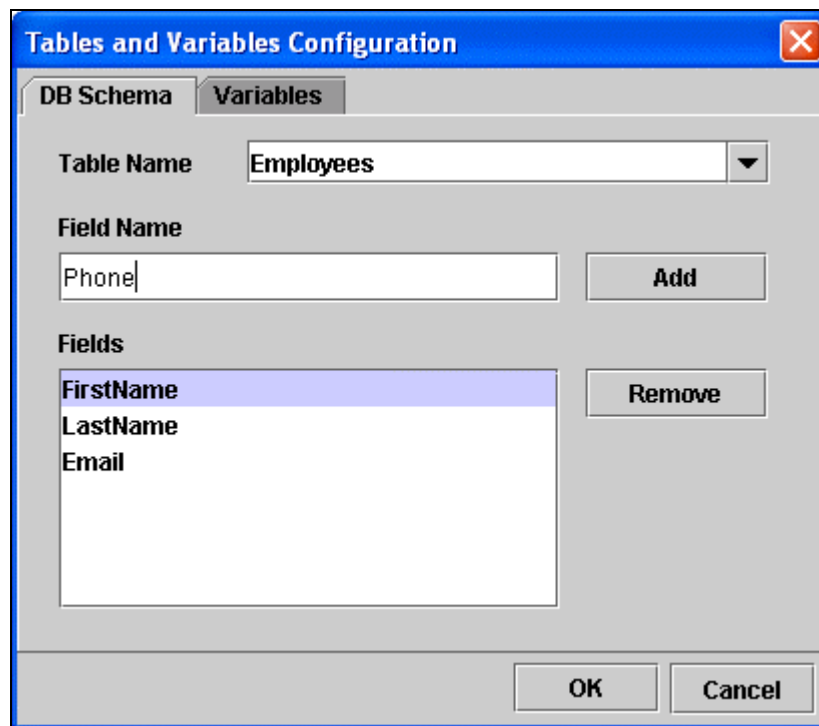


2. In the Connector Properties palette, type *Tutorial* the Data Source Name as named in the Introduction.

Complete the additional fields in the Connector Properties palette, if necessary (ex. User and Password, if required).

Next, you will need to set up the tables and fields.

3. Double-click to open the **Schema Editor**. On the DB Schema tab:
 - a. Type *Employees* in the Table Name field and press Enter.
 - b. Type *FirstName* in the Field Name field, then click Add.



- c. To add the remaining datafields, repeat Step 3b for *LastName*, *Email*, *Phone*, *Department*, and *Location*.

4. Click **OK** to close the Schema Editor.

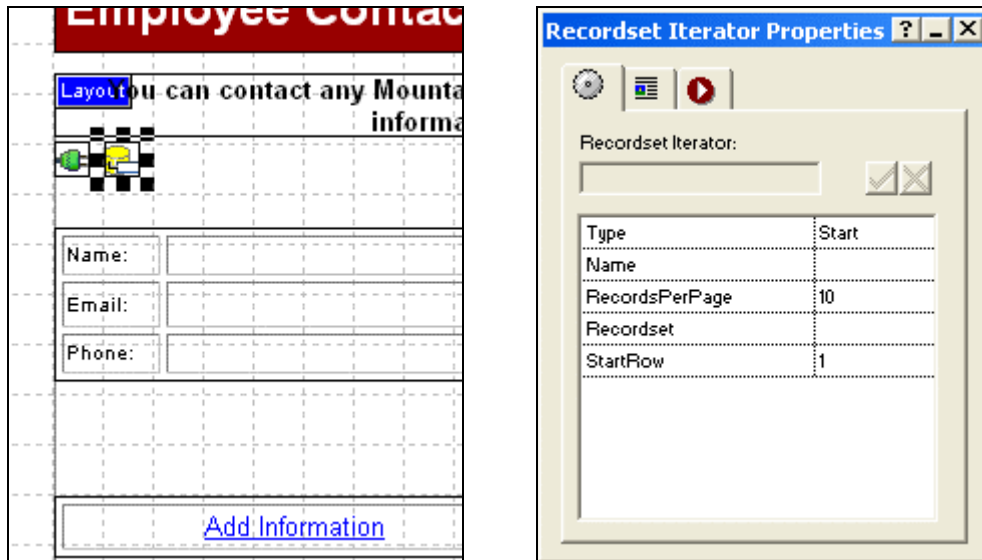
Now that you have added your table and datafields, you are ready to query the database for the employee records.

Building a Query for the Home Page

In this lesson, you will build a query that retrieves information from the Employees database to display on your Home page.

1. From the toolbar, place a **Recordset** object on the Home page. This will define the set of results the query will return.

The Recordset Object Properties Palette will appear.

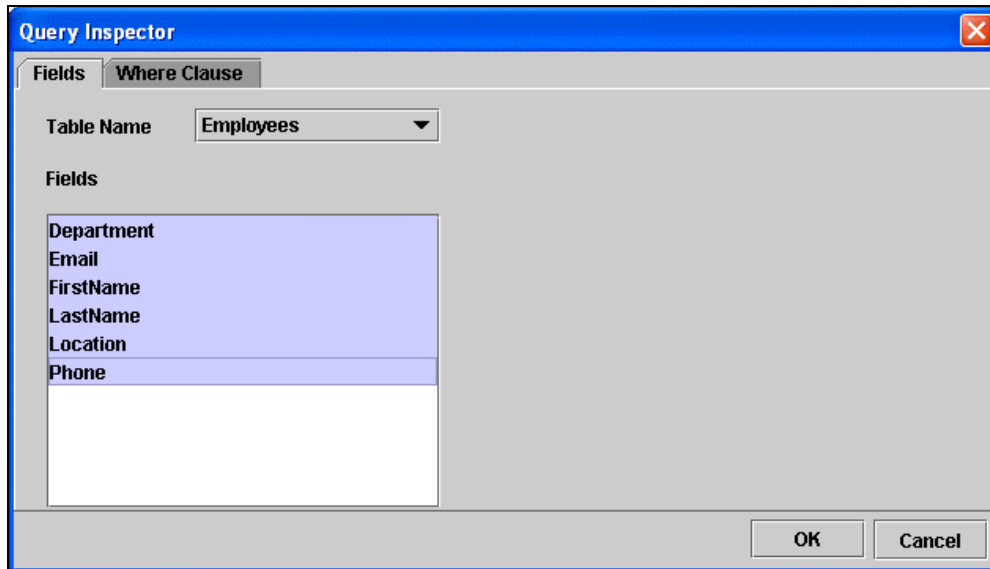


2. Type *EmployeeList* in the Name field. This will be the name of your Recordset.

Next, you will define your Recordset by selecting the table and fields that will be queried.

3. Double-click to open the SQL Wizard.
4. Select the *Employees* table from the "Table Name" drop-down list.

You will see the *Employees* datafields listed in the Fields section.



5. Highlight all datafields listed. If a datafield(s) is not highlighted, the field will not be queried and no data will be returned.
6. Click OK to close the SQL Wizard.

Now, you can determine how your Recordset results will be returned.

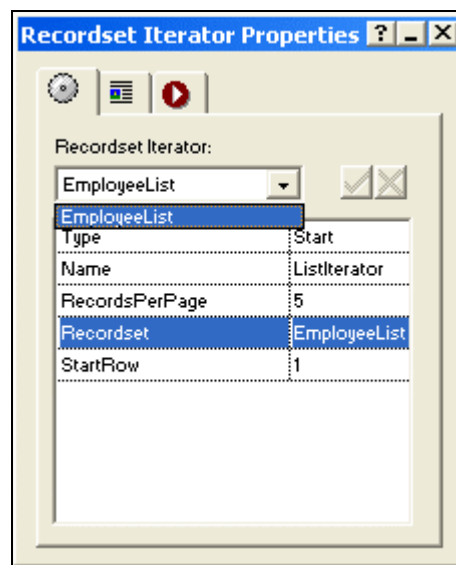
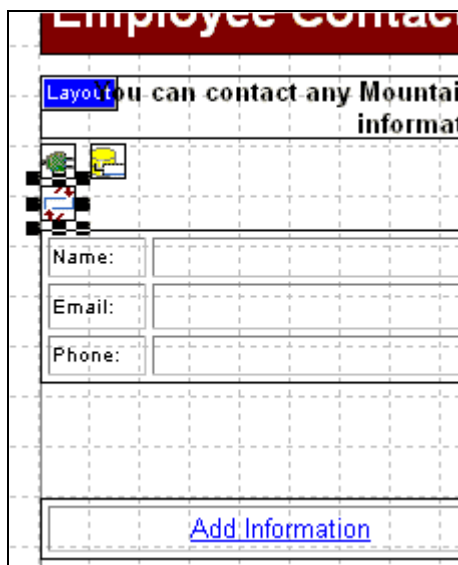
7. Type *LastName* in the **OrderBy** field. This will use the LastName datafield to order the records when displayed.
8. Choose *Ascending* in the **Sort** field to display the employee information, after ordered by LastName, in ascending (A-->Z) order.

Next, you will add objects that will iterate through your recordset and display the query results.

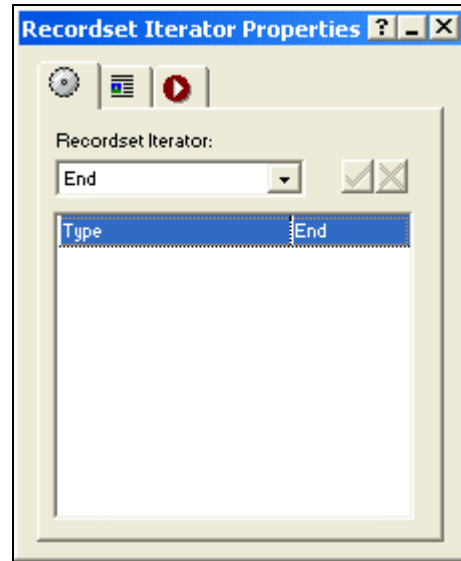
Looping through a Recordset

In this lesson, you will add Iterator objects to loop through your Recordset and return data. Note that when you use Recordset Iterator objects, you need to add at least 2 on a page - the first to begin the loop and the second to terminate it. If you do not add a terminating iterator, you will receive errors when publishing.

1. Place a **Recordset Iterator** object on the Home page. This will establish the recordset parameters.



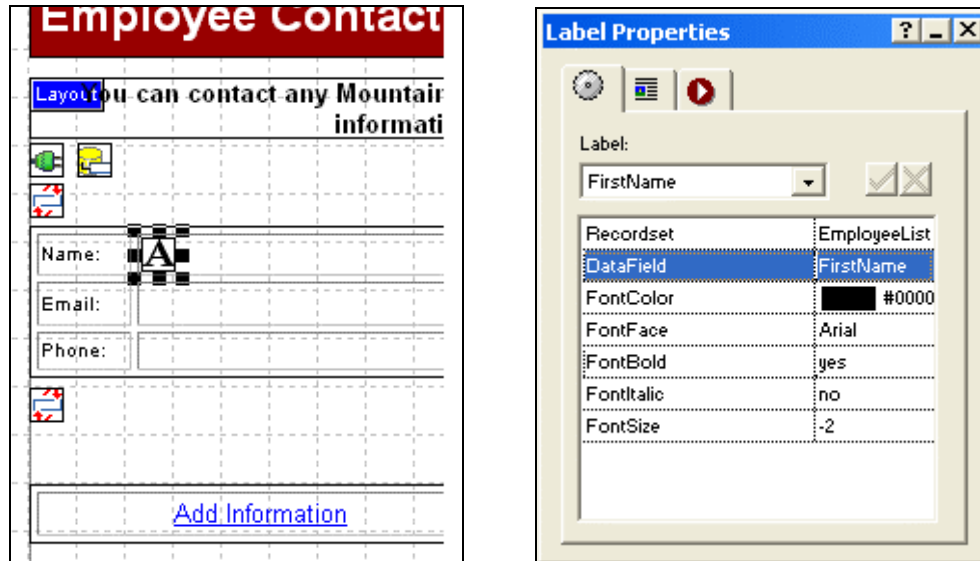
2. In the Recordset Iterator Properties Editor:
 - a. Select *Start* as the object **Type**.
 - b. Type *ListIterator* in the **Name** field. This will be the name of your Iterator.
 - c. Type 5 in the **RecordsPerPage** field. Your contact list will display 5 records at a time.
 - d. In the **Recordset** field drop-down, select *contacts* as the recordset you will be working with.
3. Place an additional Recordset Iterator object below the table and select *End* as the object **Type**.



Displaying Database Records on the Home Page

Now that you have built your query, you are ready to display the results. For this tutorial, you will use Label objects to display the recordset.

1. In the table cell next to Name, add a Label object.



2. In the Label Properties Editor:
 - a. Select *EmployeeList* from the **Recordset** field drop-down.
 - b. Select *FirstName* from the **DataField** field drop-down. This will be the field displayed.
3. Add a second Label Properties Editor next to the first, with a space between the two labels.
 - a. Select *Employees* from the **Recordset** field drop-down.
 - b. Select *LastName* from the **DataField** field drop-down. This will be the field displayed next to FirstName.
4. Add the remaining Label objects to the table cells on the page, repeating the above steps to display the appropriate fields.

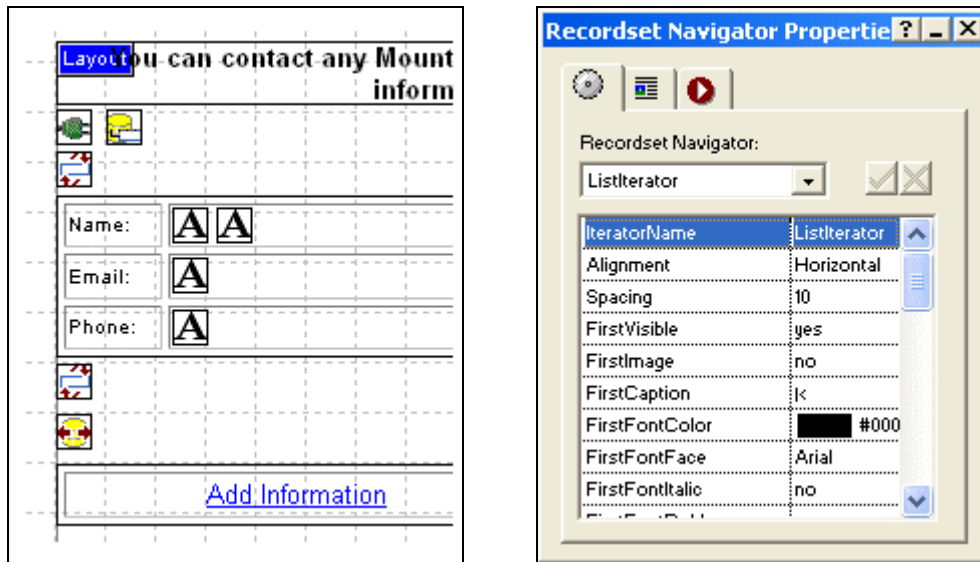
The image shows a screenshot of a web form with a dashed grid background. The form is contained within a rectangular border. At the top-left and bottom-left corners of the form's container, there are small icons of a red and blue arrow pointing in opposite directions. The form itself consists of five input fields arranged in three rows. The first row contains a single wide field labeled "Name:" with two "A" characters. The second row contains two fields: "Email:" on the left with one "A" character, and "Department:" on the right with one "A" character. The third row contains two fields: "Phone:" on the left with one "A" character, and "Location:" on the right with one "A" character.

Next, you will add navigation so that site visitors can navigate through the recordset.

Navigating through a Recordset

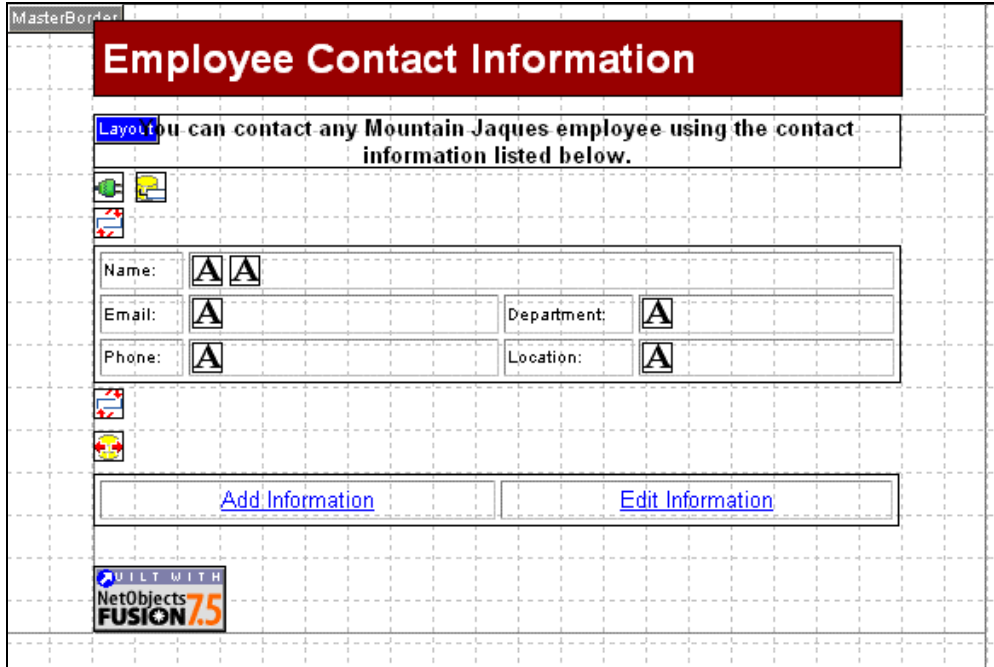
When we built the query, we set the RecordsPerPage as 5; now, we must provide your site visitors a way to navigate backwards and forwards through a recordset that contains more than 5 records. In this lesson, we will add a recordset navigator object to give visitors that functionality.

1. Place a **Recordset Navigator** object under the second iterator on the Home page .



2. In the Recordset Navigator Properties Editor, select *ListIterator* from the **IteratorName** field drop-down.

If you have completed all the previous steps in Lesson 1, your page will look like this:



You are now ready to publish your site and view the results!

Viewing the Contact List

After you have published your site, your Home page should display the Recordset results.

Employee Contact Information			
You can contact any Mountain Jaques employee using the contact information listed below.			
Name:	Jane Bennett		
Email:	jbennett@mjsports.com	Department:	Human Resources
Phone:	407-555-5678	Location:	Orlando
Name:	Ted Davis		
Email:	ted@@mjsports.com	Department:	Customer Service
Phone:	305-777-5678	Location:	Miami
Name:	Lisa Jackson		
Email:	ljackson@mjsports.com	Department:	Marketing
Phone:	831-999-5678	Location:	Tampa
Name:	Bill Jones		
Email:	bjones@mjsports.com	Department:	Marketing
Phone:	407-555-1234	Location:	Orlando
Name:	Paula Moore		
Email:	pmoore@mjsports.com	Department:	Customer Service
Phone:	904-444-5678	Location:	Jacksonville
 < < > >			
Add Information		Edit Information	

In the next lesson, you will learn how to insert records into your database using a form.

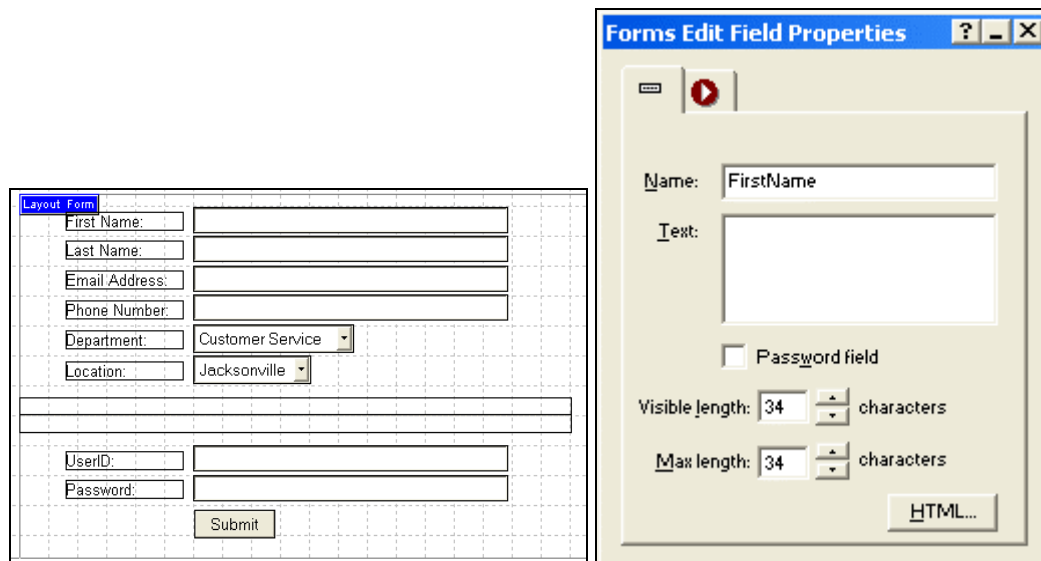
Lesson 2. Inserting Data

Using a Form to Submit New Data

The Database Component Tutorial template has pre-built forms you can use to insert and update records into your database. To complete the following lessons, you should already be familiar with the basic NetObjects Fusion concepts including creating forms.

First, we will take a look at how to use a form to submit new data. From the Site Navigation Palette, go to the **addInfo** Page. You will notice the following:

- The page layout has been changed to a Form.
- The Forms Edit Fields and Combo Boxes are named for the associated Employees database fields.

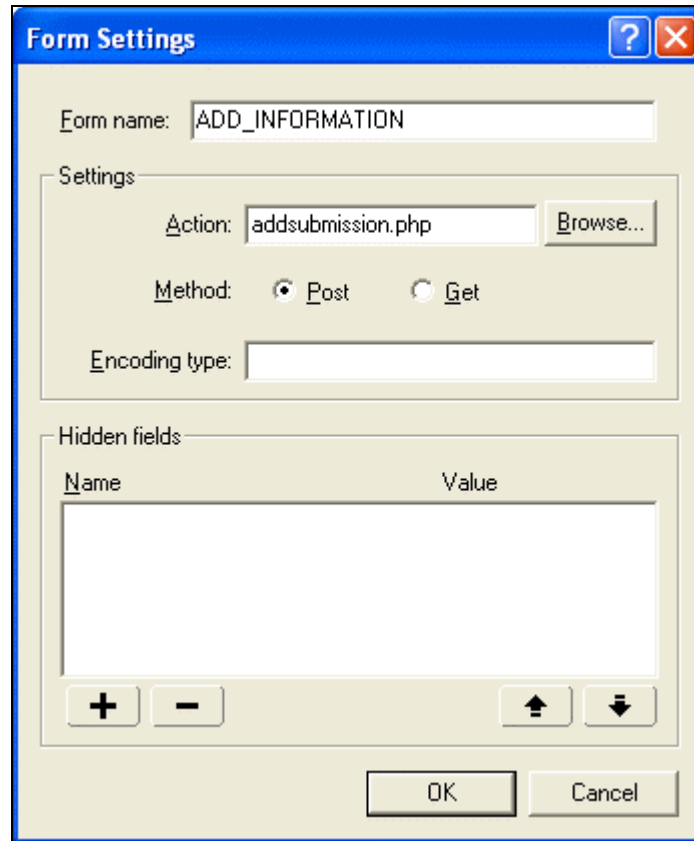


To complete the Form setup, you will need to specify the action taken after your site visitor clicks "Submit".

1. On the Add Information Page, click on "Layout Form".

The Layout Properties Palette will appear.

2. Click on the Settings... button to open the Form Settings dialog.



The image shows a 'Form Settings' dialog box with a blue title bar containing a help icon and a close icon. The dialog is divided into two main sections: 'Settings' and 'Hidden fields'. In the 'Settings' section, there is a text box for 'Form name' containing 'ADD_INFORMATION'. Below it, the 'Action' is set to 'addsubmission.php' with a 'Browse...' button. The 'Method' is set to 'Post' with radio buttons for 'Post' and 'Get'. The 'Encoding type' is an empty text box. The 'Hidden fields' section contains a table with two columns: 'Name' and 'Value'. The table is currently empty. Below the table are four buttons: a plus sign (+), a minus sign (-), an up arrow (↑), and a down arrow (↓). At the bottom of the dialog are 'OK' and 'Cancel' buttons.

3. Under Settings, add one of the following:

- If using ASP, type in addsubmission.asp
- If using ColdFusion, type in addsubmission.cfm
- If using PHP, type in addsubmission.php

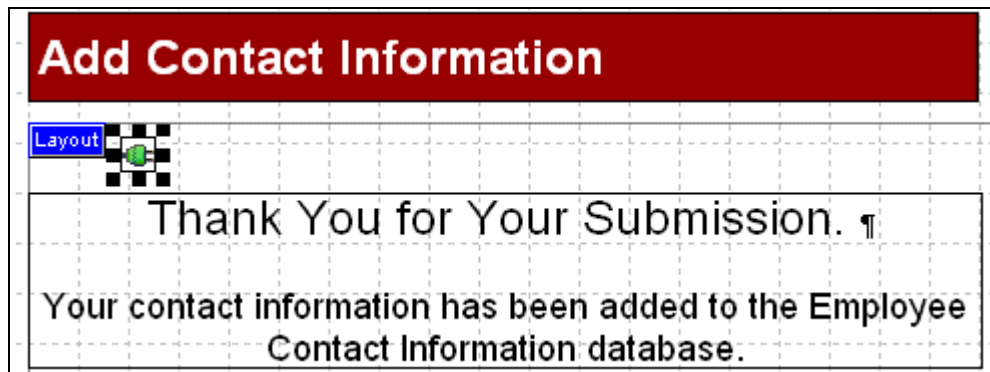
4. Click OK to close the Form Settings dialog.

Your form is complete! Next, you will build the page the new data will be submitted to.

Inserting New Records

In this lesson, you will build the page that accepts the new data and tells it what table and fields to place the data in.

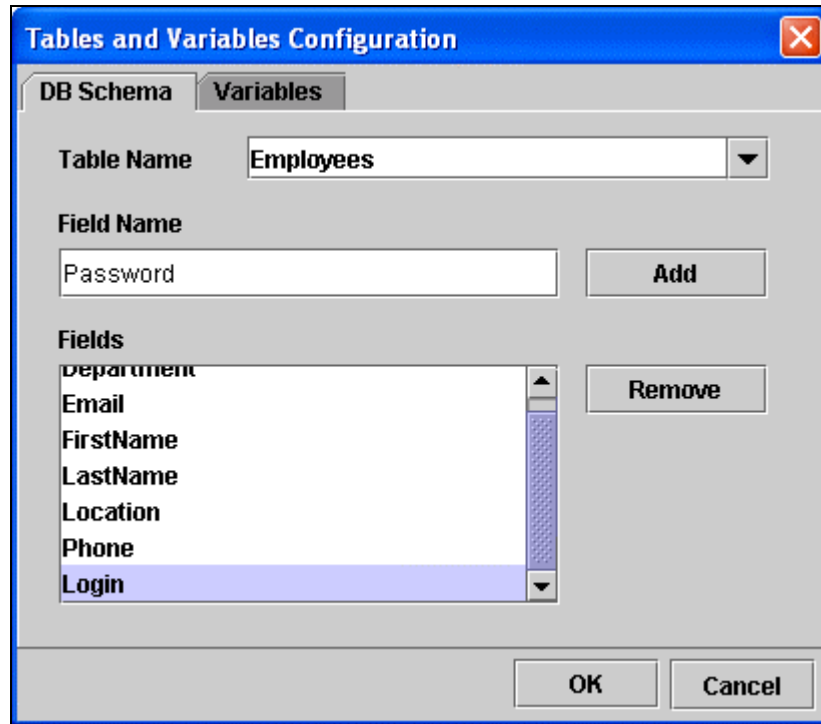
1. From the toolbar, place a Connector object on the addsubmission page and complete the required fields. This will establish a connection to the *Employees* database so that new records can be inserted.



Next, you will need to add the Login and Password datafields from the Employees database to the DB Schema.

2. In the Connector Object Properties Palette, double-click to open the **Schema Editor**.
3. On the DB Schema tab, select *Employees* from the Table Name drop-down list.

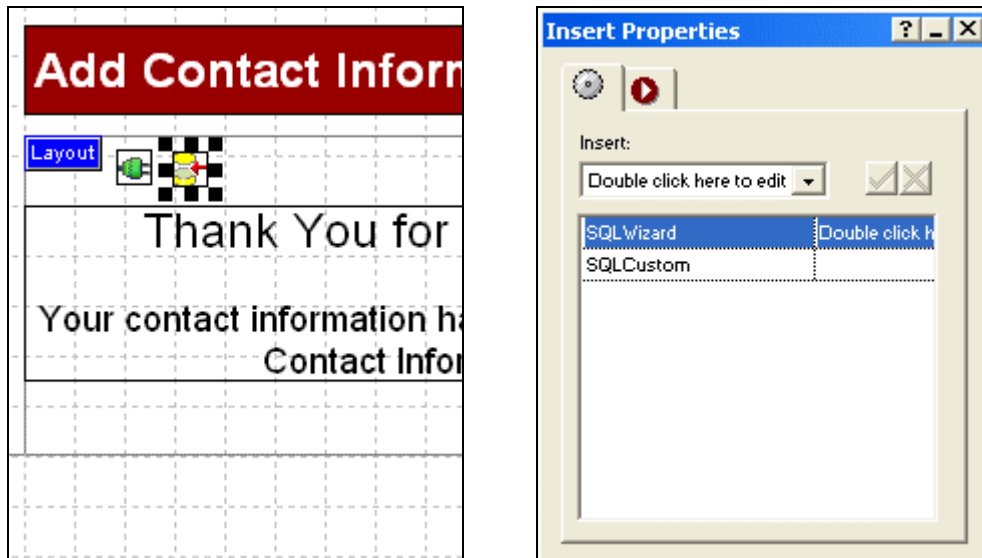
Add the datafields *Login* and *Password*.



4. Click **OK** to close the Schema Editor.

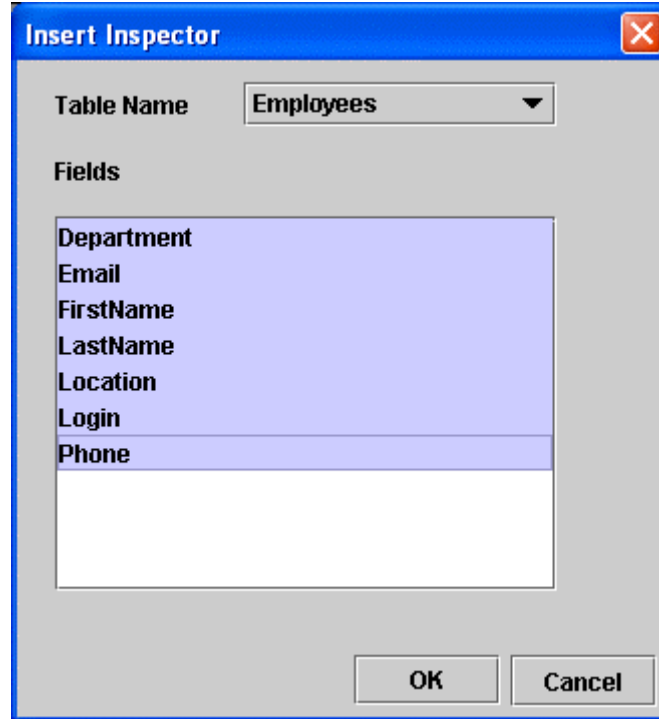
Now, you can define the table and datafields the form will submit the new information to.

5. Add an Insert object besides the Connector object on the addsubmission page.



6. In the Insert Properties palette, double-click to open the SQL Wizard.

7. Select the *Employees* table from the "Table Name" drop-down list.



8. Highlight all datafields listed. If a datafield(s) is not highlighted, the field will not be recognized and no data will be submitted to it.
9. Click OK to close the SQL Wizard.

Now, you are ready to publish and add new records!

Viewing New Records

Once you have published your site, you can add new records. To see how this works, you can add a new record on the Add Contact page.

After you click Submit....

Add Contact Information

First Name:

Last Name:

Email Address:

Phone Number:

Department:

Location:

UserID:

Password:

...go to the Home page and view the newly inserted record!

Employee Contact Information

You can contact any Mountain Jaques employee using the contact information listed below.

Name:	Paula Moore		
Email:	pmoore@mjsports.com	Department:	Customer Service
Phone:	904-444-5678	Location:	Jacksonville

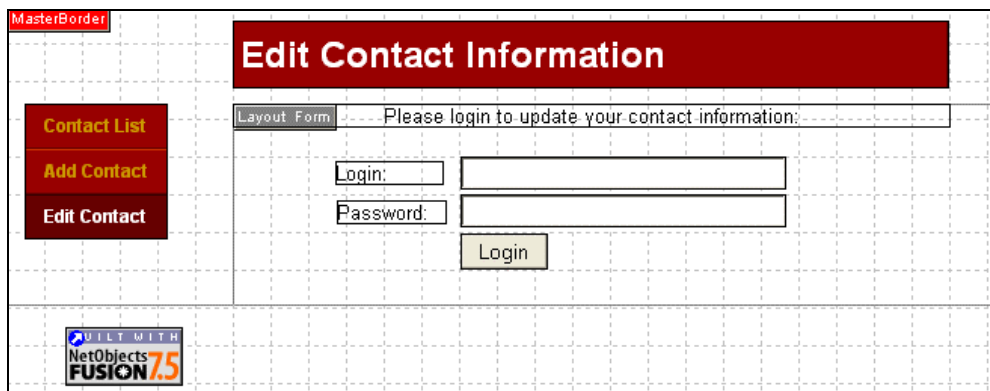
Name:	New Person		
Email:	name@mjsports.com	Department:	Customer Service
Phone:	904-555-9874	Location:	Jacksonville

Name:	John Smith		
Email:	jsmith@mjsports.com	Department:	IT

Lesson 3. Updating Data

Using a Form to Update Data

You have decided to create pages where users login, and then modify their contact information. In the following lessons, you will edit the form where the users login, then use conditional statements to determine the correct fields to display.



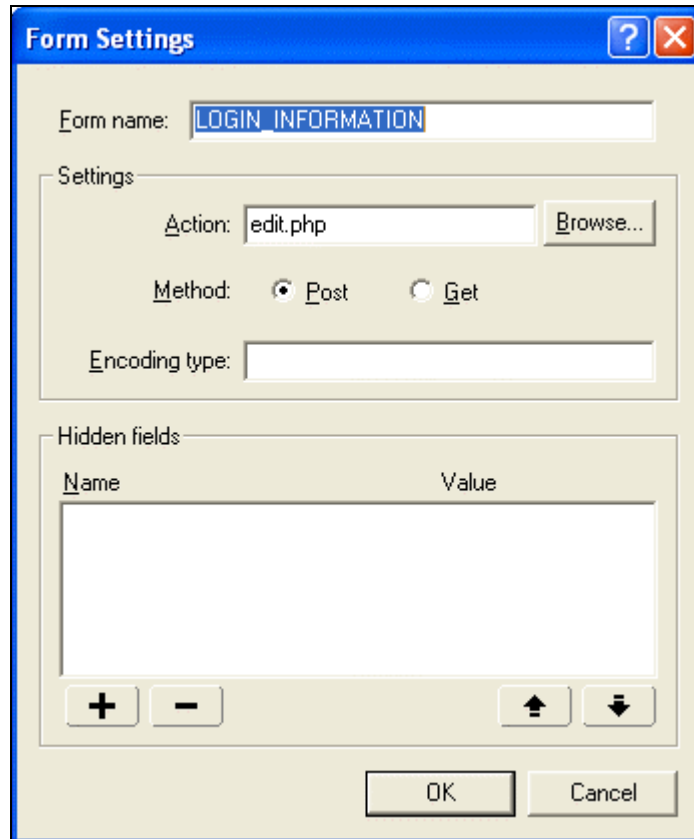
The screenshot shows a web application interface with a grid background. At the top left, there is a red box labeled "MasterBorder". Below it, a vertical sidebar contains three red buttons: "Contact List", "Add Contact", and "Edit Contact". The main content area has a red header bar with the text "Edit Contact Information". Below the header, there is a "Layout Form" containing a message: "Please login to update your contact information:". Underneath the message are two input fields: "Login:" and "Password:". Below the "Password:" field is a "Login" button. At the bottom left of the grid, there is a logo for "ULTIMATE NetObjects FUSION 7.5".

As with the previous lesson in which you learned to insert new records, you will need to complete the form setup by the action taken after your site visitor clicks "Login".

1. On the Login Page, click on "Layout Form".

The Layout Properties palette will appear.

2. Click on the **Settings...** button to open the Form Settings dialog.



The image shows a 'Form Settings' dialog box with a blue title bar. The 'Form name' field contains 'LOGIN INFORMATION'. The 'Settings' section includes an 'Action' field with 'edit.php' and a 'Browse...' button. The 'Method' section has radio buttons for 'Post' (selected) and 'Get'. The 'Encoding type' field is empty. The 'Hidden fields' section is a table with columns 'Name' and 'Value', currently empty. Below the table are buttons for adding (+) and removing (-) fields, and up/down arrow buttons. At the bottom are 'OK' and 'Cancel' buttons.

Name	Value
------	-------

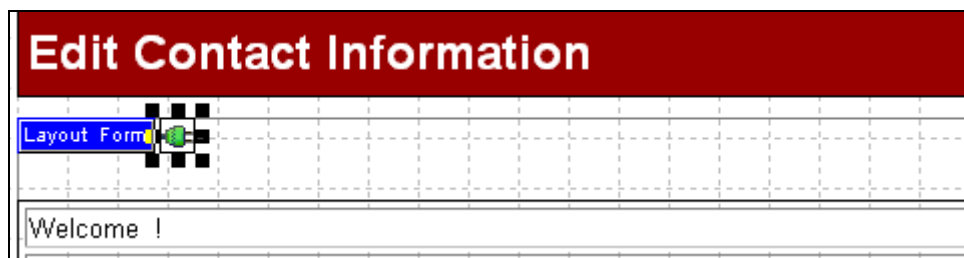
3. Under Settings, add one of the following:
 - If using ASP, type in edit.asp
 - If using ColdFusion, type in edit.cfm
 - If using PHP, type in edit.php
4. Click OK to close the Form Settings dialog.

Next, you will build the page the data to be modified is displayed on.

Using Variables and Where Clauses

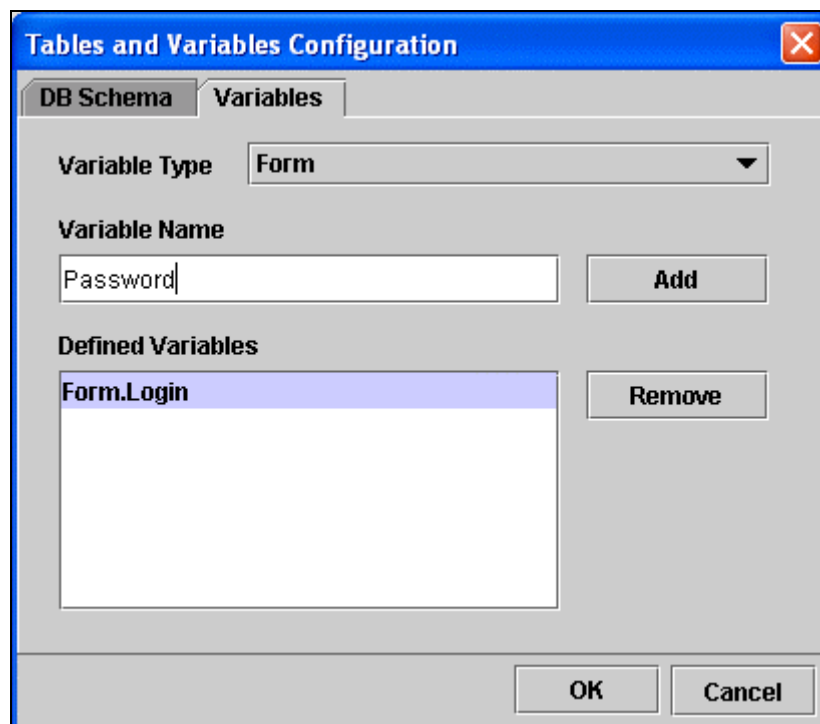
In the next 2 lessons, you will build the page that displays fields to be updated if the login information is correct. You will also add a label to display a message if the login information is incorrect.

1. From the toolbar, place a Connector object on the edit page and complete the required fields. This will establish a connection to the *Employees* database.



2. In the Connector Object Properties Palette, double-click to open the **Schema Editor**.
3. On the Variables tab, select *Form* from the Variable Type drop-down list.

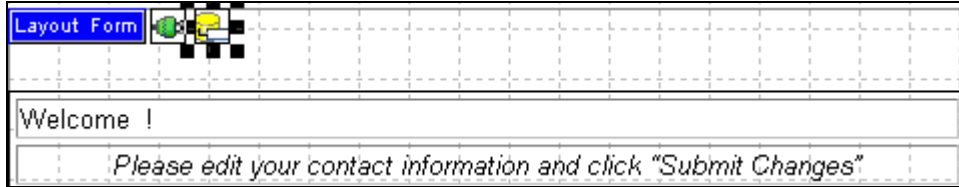
Add the form variables *Login* and *Password*.



4. Click **OK** to close the Schema Editor.

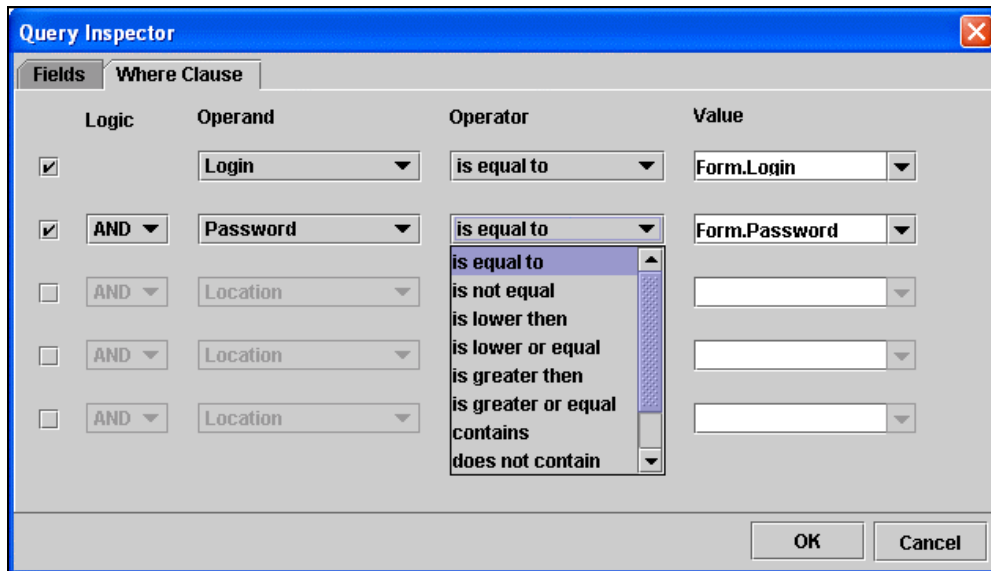
Now, you can define the table and datafields the form will retrieve information from.

5. Add a recordset object next to the Connector object on the edit page.



In the Recordset Properties palette, name your recordset *updates*.

6. Double-click to open the **SQL Wizard**.
 - a. In the Table Name drop-down, select the Employee table.
 - b. Highlight the data fields listed.
 - c. On the Where Clause tab, set the following:
Login is equal to Form.Login AND Password is equal to Form.Password.



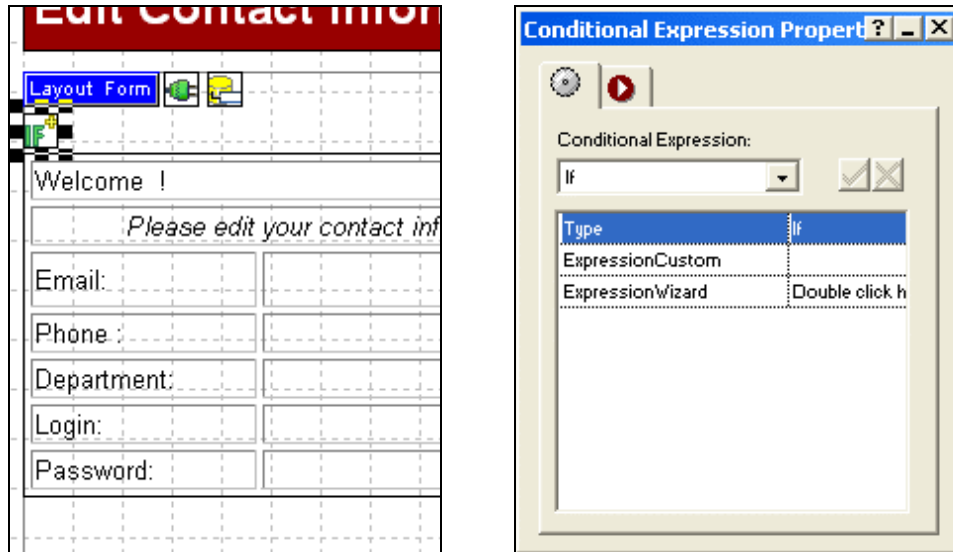
7. Click **OK** to close the SQL Wizard.

Now that you have set up verification and will return the correct record associated with a login and password, you are ready to add conditions to the query.

Adding Conditional Expressions

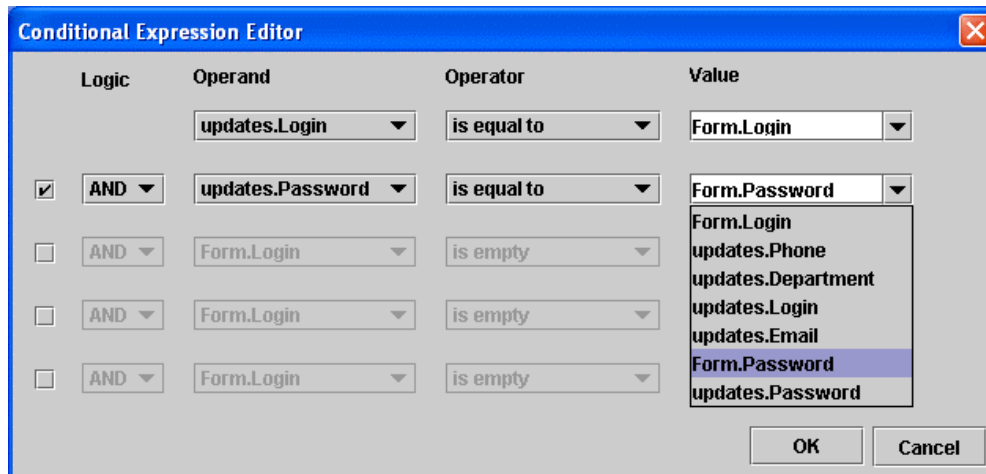
In this lesson, you will add conditions that will determine which fields to display. If the login and password match a record in the database, the site visitor will have the ability to modify his contact information. If not, a message asking the user to try again will be displayed.

1. Add an If (Conditional) object above the table on the edit page.



2. In the If (Conditional) Properties palette, double-click to open the Expression Wizard and set the following:

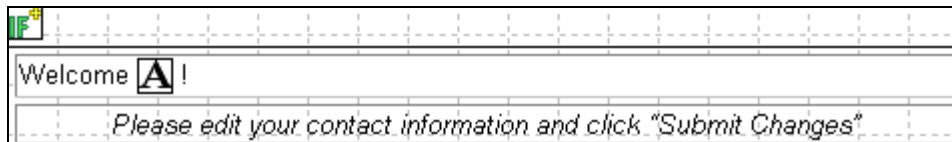
Login is equal to Form.Login AND Password is equal to Form.Password.



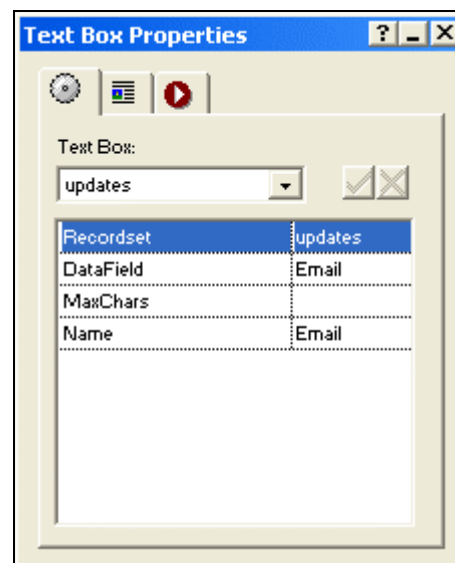
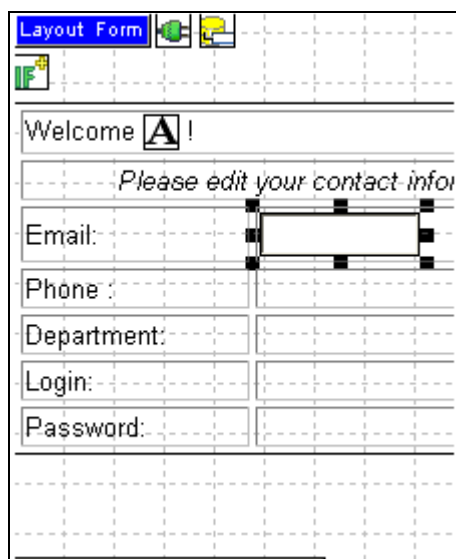
3. Click **OK** to close the Expression Editor.

Next, you will add a label to identify the user and text fields to display data that can be edited.

4. In the first table cell, add a Label object between "Welcome" and "!".



5. In the table cell next to "Email", add a Textbox object.



6. In the Text Box Properties palette:

Website Pros Database Component

- a. Select *updates* from the Recordset drop-down list.
- b. Select *Email* as the datafield you will be displaying.
- c. Type Email in the Name field.

7. Add Textbox objects in the cells next to "Phone", "Department", "Login", and "Password".


Email:	<input type="text"/>
Phone :	<input type="text"/>
Department:	<input type="text"/>
Login:	<input type="text"/>
Password:	<input type="text"/>

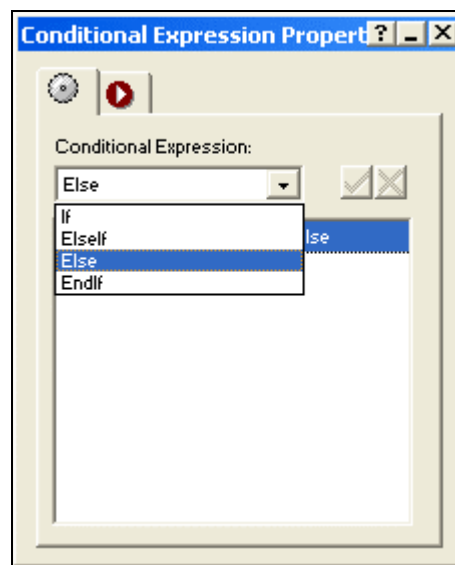
Repeat step 3 for each, selecting the appropriate datafield and typing in the matching name.

Now, you will add text to display if the first condition is not met.

8. Add an additional If (Conditional) object to the page under the "Submit Changes" button.

In the If (Conditional) Properties palette, select Else as the type.

Please edit your contact	
Email:	<input type="text"/>
Phone :	<input type="text"/>
Department:	<input type="text"/>
Login:	<input type="text"/>
Password:	<input type="text"/>
<input type="button" value="Submit Changes"/>	
	Your login



Now, if the first condition is not met, a message will display asking the user to "Try Again".

9. Add a final If (conditional) object to the page to terminate the loop.

In the If (Conditional) Properties palette, select EndIf as the type.

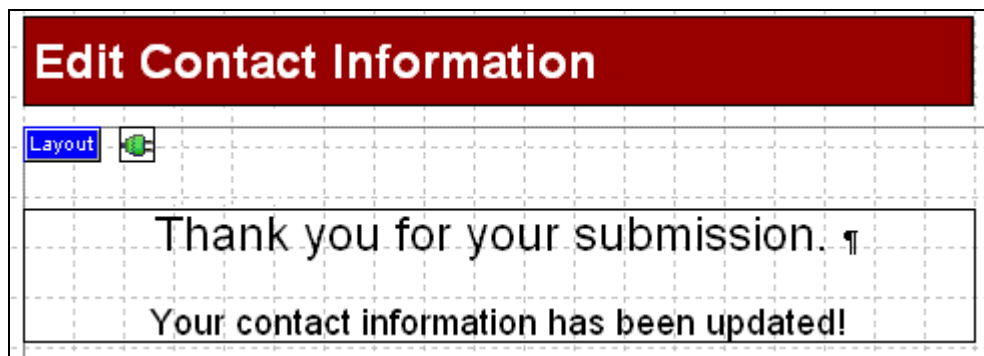
If you have completed all the previous steps in Lesson 3, your page will look like this:

The screenshot shows a web form titled "Edit Contact Information" with a red header. Below the header is a "Layout Form" toolbar. The form contains a "Welcome A!" message, a prompt to edit contact information, and five input fields for "Email:", "Phone:", "Department:", "Login:", and "Password:". A "Submit Changes" button is located below the input fields. At the bottom, a message box displays "Your login information is incorrect. ← Please try again!". The form is overlaid on a grid background.

Updating Records

In this lesson, you will add an object to the page that specifies what table and fields to update.

1. From the toolbar, place a Connector object on the editsubmission page. This will establish a connection to the *employees* database so that updated records can be submitted.

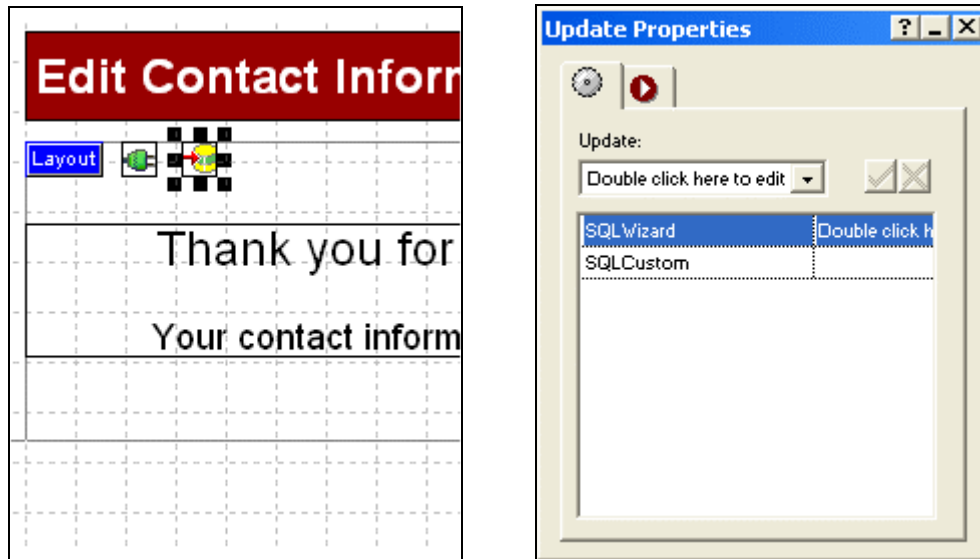


Next, you will need to add additional form variables to the DB Schema. These will be used to specify the form fields that will update the datafields.

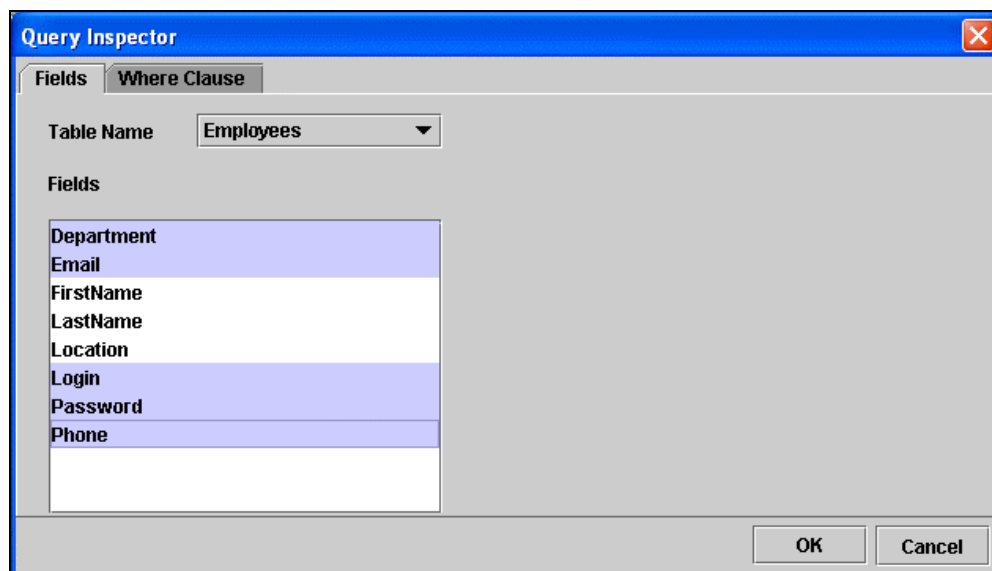
2. In the Connector Object Properties Palette, double-click to open the Schema Editor.

Now, you can define the table and datafields that will be updated using a Where Clause.

5. Add an Update object besides the Connector object on the editsubmission page.



6. In the Update Properties palette, double-click to open the SQL Wizard.
 - a. Click on the Fields tab and highlight the Department, Email, Login, Password, and Phone fields.



- b. Click on the Where Clause tab and set the following:

Login *is equal to* *Form.Login*
And Password *is equal to* *Form.Password*

Website Pros Database Component

7. Click **OK** to close the Schema Editor.

Now, you are ready to publish and update records!

Viewing Updated Records

Once you have published your site, you can modify records. To see how this works, you can Login...

...edit existing data....

...and go to the Home page to view the updated record!

Employee Contact Information

You can contact any Mountain Jaques employee using the contact information listed below.

Name:	Jane Bennett		
Email:	jane.bennett@mjsports.com	Department:	Human Resources
Phone:	407-555-5678	Location:	Orlando

Name:	Ted Davis		
Email:	ted@@mjsports.com	Department:	Customer Service
Phone:	305-777-5678	Location:	Miami

Getting Help

Connecting to a database or table.

1. Test your connection to see that it is working.
2. Verify your server is configured correctly.
3. Use the Schema Editor in the Connector Properties palette to add tables and fields.
4. The connector does not automatically connect to your database and populate this information; tables and fields must be added manually.
5. If using a custom engine (PHP only), verify the correct driver has been installed in the Components>Connector folder. See Before Getting Started.
6. If case sensitive (Unix), verify your tables and fields are entered correctly.

Displaying records.

1. If using conditional objects (If-Else-ElseIf), verify that you have added a closing (Type = EndIf) tag to the page.
2. If using a recordset iterator object, verify that you have added a closing (Type= Close) tag to the page.
3. If using a where clause, verify that your variables and settings are correct.
4. If using recordset iterator and/or conditional objects, verify the object placement is correct. For example, verify that your beginning navigator object is placed above the closing object so the code is generated correctly.

Modifying and deleting records.

1. Use a where clause to filter records so that the correct table/fields are updated and/or deleted.
2. If using a where clause, verify that your variables and settings are correct.

Navigating through recordsets.

1. If using a recordset iterator object, verify that you have added a closing (Type= Close) tag to the page.
2. If using a form variable, verify the method type (in the Form Settings Properties) is set to "Get".
3. Verify the MaxRows and StartRow attributes are correct.

Using Forms.

1. If using a form variable, verify the method type (in the Form Settings Properties) is set to "Get".
2. Verify the form field names are identical to the field names of the table you are using.
3. Verify the Form Settings properties contains the correct path to and extension of the submission page.

Website Pros Database Component

For issues with the Database Component, please contact NetObjects Technical Support.

- support@netobjects.com

For server side issues, please check with your hosting provider. You can also visit the following links for additional information:

- ColdFusion: <http://www.macromedia.com/software/coldfusion/productinfo/systemreqs/>
- ASP: http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnasp/html/msdn_aspfaq.asp
- PHP: <http://www.php.net/>
- MySQL: <http://www.mysql.com/>

Index

A

Add

- Component Objects 4
- Conditions 15
- Data Source 8
- Database Component toolbar 3
- Tables 12
- Toolbar 3

Add 3, 4, 8, 9, 10, 12, 15

Add SQL 18, 19, 23

Add Table 13

Added/placed 4

ALT 20

And/or 1, 5

Ascending 18

ASP 1, 3, 59

Attribute's

enter 5

Attribute's 5

B

Before Getting Started 1

Biology 11, 14

Bolds

FirstCaption 24

LastCaption 24

NextCaption 24

PreviousCaption 24

Bolds 24

BORDER 20

Build

recordset 8

Build 8

C

CACHE DURATION 18

CAPTION 21

CGI

reflecting 8

CGI 8

CGI Environment 8

Checkbox 13, 15

Classification 11, 14

Classification datafield
searching 14

Classification datafield 14

ColdFusion 1, 3, 59

Combo

name 22

Combo 22

ComboBox

name 22

ComboBox 22

Component Object Definitions

see 5

Component Object Definitions 5

Component Objects

Adding 4

Component Objects 4

Component Toolbar 3

Components Tools 3

Conditional Expressions Editor 15

Conditions

Adding 15

Conditions 15

Configuring

Tables 9

Variables 10

Configuring 9, 10

Connector 8, 17

Connector Object 3

Control Panel 1

CONTROL STYLE 22

CTRL 12

D

DATA FIELD 20, 21, 22, 23

Data Output 18

Data Source

add 8

Data Source 8

Database

Select 3

Database 3

Database Component 1, 7

Database Component Toolbar

add 3

indicates 3

Database Component Toolbar 3, 4

Website Pros Database Component

Database Component Works 3

Datafield

 select 5

Datafield 5

Datafields 11, 12

DATASOURCE 17

DB Language Name 3

DB Schema 8, 9

Defined 16

Defined Variables list 10

Delete 11, 12, 13, 19

Displaying

 recordSet 22

Displaying 22

DSN 17, 59

E

Editor

 use 5

Editor 5, 7

Else 23

ElseIf

 including 14

ElseIf 14, 23

ElseIf Classification 14

EndIf 14, 23

Enter

 attribute's 5

Enter 5

EXECUTION TIMEOUT 18

EXPRESSION CUSTOM 23

Expression Wizard 8, 14, 23

ExpressionWizard 14

F

Field Name 9

Fields 11, 12

Fields list 9, 12

Filtering

 Results 13

Filtering 13

First Name 1

FirstCaption

 Bolds 24

FirstCaption 24

FirstFontBold 24

FirstFontColor 24

FirstFontFace 24

FirstFontItalic 24

FirstFontSize 24

FirstImage 24

FirstImageSrc 24

FirstName 1

FirstVisible 24

FONT BOLD 20, 21

FONT COLOR 20, 21

FONT FACE 20, 21

FONT ITALIC 20, 21

FONT SIZE 20, 21

FontSize 5

H

HEIGHT 20

Hidden Field

 name 23

Hidden Field 23

HiddenField 23

Highlighted 59

Hyperlink 21

I

IF 23

If-Else-ElseIf-EndIf 5

IMAGE 20, 21

IMAGE SRC 21

Including

 ElseIf 14

Including 14

Indicates

 Database Component toolbar 3

Indicates 3

Insert 11, 12, 13, 19

Iterator

 Name 24

Iterator 4, 24, 59

Iterator Name 24

IteratorName 5

L

Label 20

LastCaption

 Bolds 24

LastCaption 24

LastFontBold 24

LastFontColor 24

LastFontFace 24

LastFontItalic 24

- LastFontSize 24
- LastImage 24
- LastImageSrc 24
- LastName 11
- LastVisible 24
- LINES 22
- Link 21
- ListBox 22
- M**
- Major 11
- Major datafield 11
- MAX CHARS 21
- MAXROWS 18
- Microsoft Access 1
- N**
- NAME
 - Combo 22
 - ComboBox 22
 - Hidden Field 23
 - iterator 24
- NAME 5, 18, 21, 22, 23, 24
- Needing
 - start/close 5
- Needing 5
- NetObjects Fusion 1, 5
- NextCaption
 - Bolds 24
- NextCaption 24
- NextFontBold 24
- NextFontColor 24
- NextFontFace 24
- NextFontItalic 24
- NextFontSize 24
- NextImage 24
- NextImageSrc 24
- NextVisible 24
- Not Defined 16
- O**
- Object Attributes
 - Setting 5
- Object Attributes 5
- OK 9, 10, 12, 13, 15
- Online 1
- Operand 13, 15
- Operator Definitions 16
- ORDER BY 18
- P**
- Page 3, 4, 5
- Password 17
- PHP 1, 59
- PHP application 3
- PreviousCaption
 - Bolds 24
- PreviousCaption 24
- PreviousFontBold 24
- PreviousFontColor 24
- PreviousFontFace 24
- PreviousFontItalic 24
- PreviousFontSize 24
- PreviousImage 24
- PreviousImageSrc 24
- PreviousVisible 24
- Properties 4, 5
- Properties Palette 3, 4, 5
- Providing
 - textbox field 21
- Providing 21
- Q**
- Query Inspector 12, 13
- R**
- RECORD SET 18, 20, 21, 22, 23, 24
- Record Set Iterator 24
- Record Set Navigator 24
- RECORDS PER PAGE 24
- Records Won't Display 59
- Recordset
 - build 8
 - Displays 22
 - requires 5
- Recordset 5, 8, 11, 14, 22
- Recordset Iterator
 - requires 5
- Recordset Iterator 4, 5
- Recordset Navigator 59
- Recordset Properties Palette 8
- Reflecting
 - CGI 8
- Reflecting 8
- Repeat Steps 9, 10
- Requires
 - recordset 5
 - recordset iterator 5

Website Pros Database Component

Requires 5

Results

Filtering 13

Results 13

S

S 17, 24

SCHEMA EDITOR

Using 8

SCHEMA EDITOR 8, 9, 10, 17

SchemaEditor 8, 10, 11

Searching

Classification datafield 14

Searching 14

See

Component Object Definitions 5

See 5

Select

Database 3

datafield 5

Select 3, 5

Setting

Object Attributes 5

Setting 5

SORT 18

SQL 3, 7, 8

SQL CUSTOM 18, 19

SQL Editor

Use 17, 18, 19, 23

SQL Editor 17, 18, 19, 23

SQL Server 1

SQL Wizard

Using 11

SQL Wizard 11, 12, 13, 18, 19, 59

Start 4

START ROW 24

Start/close

needing 5

Start/close 5

Statment 23

Step 12, 13

Students 11, 14

T

Table Name 9, 12

Tables

Adding 12

Configuring 9

Tables 9, 12

Tag Library Component 1

TextArea 22

Textbox 21

Textbox field

providing 21

Textbox field 21

Toolbar

Adding 3

Toolbars 3

Toolbars>Component Tools 3

TYPE 5, 23, 24

U

Update 11, 19

URL 8

URL Parameter 8

Use

Editor 5

SQL Editor 17, 18, 19, 23

Website Pros Database Component 1

Where 11

Where Clause 11

Use 1, 5, 11, 17, 18, 19, 23

USER 17

Username 17

Using

Schema Editor 8

SQL Wizard 11

Using 8, 11

V

V.7.5 Update 1

Value 13

Variable Name 10

Variables

Configuring 10

Variables 8, 10

View menu 3

W

Web site 1

Website Pros 1

Website Pros Database Component

use 1

Website Pros Database Component 1

Where

use 11

Where Clause

use 11
Where Clause 11, 13
WIDTH 20

Y

YOU MUST HIGHLIGHT DATA
FIELDS BEFORE CLOSING THE

EDITOR OR THEY WILL NOT
APPEAR IN OBJECT PROPERTIES
PALETTES 12