Building Advanced Queries In R:BASE eXtreme 9.5

R:BASE Technologies, Inc.

Building Advanced Queries In R:BASE eXtreme 9.5

by R:BASE Technologies, Inc.

The Query Builder is an SQL (Structured Query Language) building utility for retrieving data from R:BASE tables and views. A query is a question constructed to get information from database tables and columns. A view is basically a query saved within the database.

Building Advanced Queries In R:BASE eXtreme 9.5

Printed: January 2014 in Murrysville, PA

First Edition

Table of Contents

Part I

1 Building a Query Using a Union

A "Union" can be added to your queries in order to combine the results of two or more SELECT statements. This optional operator combines the results of two SELECT commands or clauses, displaying the results of the second SELECT command below those of the first. By default, unions deletes duplicate rows.

Include the optional keyword ALL to include duplicate rows in the final result.

The UNION operator requires the following three conditions:

- The SELECT statements must specify an equal number of columns.
- Columns that are being combined must have the same data type.
- Only the last SELECT statement can contain an ORDER BY clause.

1.1 Quarterly Summary Query

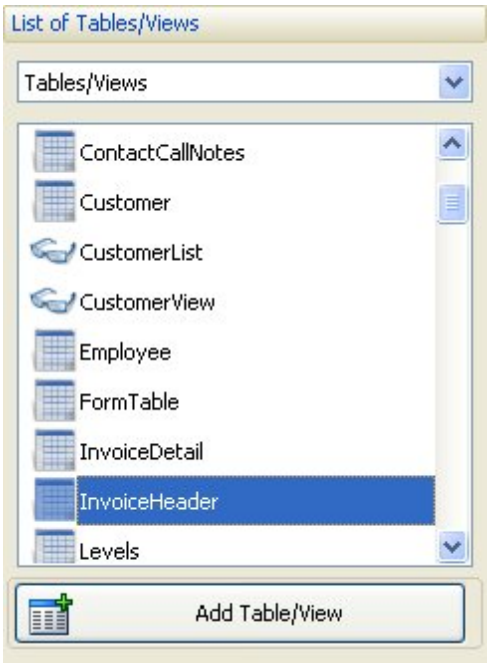
In the RRBYW18 sample database, the "QuarterlySummary" view example utilizes three UNION statements with an initial SELECT statement. The following instructions will step through the process to recreate the view in the Query Builder.

To continue, launch R:BASE eXtreme 9.5 and connect to the RRBYW18 sample database. If you are using R:BASE eXtreme 9.5 (64), the database is located in the following default installation directory: "C:\RBTI\RBG95_64\Samples\RRBYW18". If you are using R:BASE eXtreme 9.5 (32), the database is located in the following default installation directory: "C:\RBTI\RBG95_32\Samples\RRBYW18".

1. Start R:BASE eXtreme 9.5
2. Connect to the RRBYW18 sample database, by selecting "Database" > "Connect", and navigating to the above default installation directory based upon your version.
3. Then, launch the Query Builder by selecting "Tools" > "Query by Example" from the main Menu Bar

1.1.1 1st Quarter

In the Query Builder you will see a list of tables and views within a panel to the left that can be added to your query. The list contains all tables and views for the connected database. The table used in this example is InvoiceHeader.



1. Add the "InvoiceHeader" to the query by selecting the table, and then selecting the "Add Table/View" button.

   The table should now be listed under "Tables/Views In Use". With the table added, the specific columns to be used in the query must be selected.

2. Right click on the "InvoiceHeader" table under "Tables/Views In Use", and choose "Select Columns" from the speed menu.

   The "InvoiceHeader" columns will be displayed to be added to the query. This dialog window is basically a graphic representation of the SELECT command. When working with columns in your queries, expressions can also be added to the columns to perform calculations.

3. In this example, select the "New Expression" button to display the Expression Builder.
4. Enter the value "1" into the "Expression Text" panel. The value "1" is the numeric value that will represent the 1st quarter of the figures in the query. Press the "OK" button to save the expression.
5. Select the "New Expression" button to once again display the Expression Builder.
6. Within the the "Function Templates" panel, select use the drop-down combo box to select the aggregate function "SUM", which will be displayed as "(SUM(arg))"
7. Select the "Use" button to add the function to the "Expression Text" panel.
8. Place the cursor within the set of parenthesis and delete the "arg" value.
9. Then, with the cursor still in the parenthesis, use the "Select Columns" list box to locate and select the "InvoiceTotal" column.
10. Once selected, press the "Add Column" button to insert the column into the "Expression Text" panel.

    The Expression panel should have the following displayed:

    ```
    (SUM( T1.InvoiceTotal ))
    ```

11. Click the OK button to close the Expression Builder.
12. Click the OK button to close the column selection dialog and return to the Query Builder main window.
13. Right click on the "InvoiceHeader" table under "Tables/Views In Use", and choose "WHERE Clause" from the speed menu.

    The WHERE Builder will be displayed to place a condition on the view in order to limit the records displayed.

14. Use the "Select Columns" list box to locate and select the "TransDate" column.

15. Once selected, press the "Add Column" button to insert the column into the "Expression Text" panel.
16. From the "Operators and Keywords" panel, select the "BETWEEN" button to add the keyword to the WHERE Clause panel.
17. Type in the **first** quarter starting date of "01/01/2011".
18. From the "Operators and Keywords" panel, select the "AND" button to add the keyword to the WHERE Clause panel.
19. Type in the **first** quarter ending date of 03/31/2011".

The Expression panel should have the following displayed:

```
TransDate BETWEEN 01/01/2011 AND 03/31/2011
```

20. Click the OK button to close the WHERE Builder.

Back in the Query Builder main window, you should see the SQL syntax displayed in the bottom of the page as:

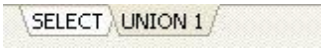
```
SELECT 1,(SUM( T1.InvoiceTotal ))
FROM  InvoiceHeader T1
WHERE TransDate BETWEEN 01/01/2011 AND 03/31/2011
```

1.1.2 2nd Quarter

To add the 2nd Quarter results, we will begin by adding a UNION to the query. A UNION can be added to queries in order to combine the results of two or more SELECT statements.

1. Select the "Add UNION" button from Query Builder Toolbar.

Take notice that a "UNION 1" tab has appeared in the Query Builder. These tabs permit switching from multiple UNIONs within a query.



After this point, the steps are very similar to building the 1st quarter statement.

2. Add the "InvoiceHeader" to the query by selecting the table, and then selecting the "Add Table/View" button.

The table should now be listed under "Tables/Views In Use". With the table added, the specific columns to be used in the query must be selected.

3. Right click on the "InvoiceHeader" table under "Tables/Views In Use", and choose "Select Columns" from the speed menu.
4. In this example, select the "New Expression" button to display the Expression Builder.
5. Enter the value "2" into the "Expression Text" panel. The value "2" is the numeric value that will represent the 2nd quarter of the figures in the query. Press the "OK" button to save the expression.
6. Select the "New Expression" button to once again display the Expression Builder.
7. Within the the "Function Templates" panel, select use the drop-down combo box to select the aggregate function "SUM", which will be displayed as "(SUM(arg))"
8. Select the "Use" button to add the function to the "Expression Text" panel.
9. Place the cursor within the set of parenthesis and delete the "arg" value.
10. Then, with the cursor still in the parenthesis, use the "Select Columns" list box to locate and select the "InvoiceTotal" column.
11. Once selected, press the "Add Column" button to insert the column into the "Expression Text" panel.

The Expression panel should have the following displayed:

```
(SUM( T1.InvoiceTotal ))
```

12. Click the OK button to close the Expression Builder.
13. Click the OK button to close the column selection dialog and return to the Query Builder main window.
14. Right click on the "InvoiceHeader" table under "Tables/Views In Use", and choose "WHERE Clause" from the speed menu.
15. Use the "Select Columns" list box to locate and select the "TransDate" column.
16. Once selected, press the "Add Column" button to insert the column into the "Expression Text" panel.
17. From the "Operators and Keywords" panel, select the "BETWEEN" button to add the keyword to the WHERE Clause panel.
18. Type in the **second** quarter starting date of "04/01/2011".
19. From the "Operators and Keywords" panel, select the "AND" button to add the keyword to the WHERE Clause panel.
20. Type in the **second** quarter ending date of 06/30/2011".

    The Expression panel should have the following displayed:

    ```
    TransDate BETWEEN 04/01/2011 AND 06/30/2011
    ```

21. Click the OK button to close the WHERE Builder.

    In the Query Builder main window, you should see the SQL syntax displayed in the bottom of the page as:

    ```
    SELECT 1,(SUM( T1.InvoiceTotal ))
    FROM  InvoiceHeader T1
    WHERE T1.TransDate BETWEEN 01/01/2011 AND 03/31/2011
    UNION SELECT 2,(SUM( T1.InvoiceTotal ))
    FROM  InvoiceHeader T1
    WHERE T1.TransDate BETWEEN 04/01/2011 AND 06/30/2011
    ```

1.1.3 3rd Quarter

To add the 3rd Quarter results, we will add another UNION to the query.

1. Select the "Add UNION" button from Query Builder Toolbar.

   Take notice that a "UNION 2" tab has appeared in the Query Builder.

2. Add the "InvoiceHeader" to the query by selecting the table, and then selecting the "Add Table/View" button.
3. Right click on the "InvoiceHeader" table under "Tables/Views In Use", and choose "Select Columns" from the speed menu.
4. In this example, select the "New Expression" button to display the Expression Builder.
5. Enter the value "3" into the "Expression Text" panel. The value "3" is the numeric value that will represent the 3rd quarter of the figures in the query. Press the "OK" button to save the expression.
6. Select the "New Expression" button to once again display the Expression Builder.
7. Within the the "Function Templates" panel, select use the drop-down combo box to select the aggregate function "SUM", which will be displayed as "(SUM(arg))"
8. Select the "Use" button to add the function to the "Expression Text" panel.
9. Place the cursor within the set of parenthesis and delete the "arg" value.
10. Then, with the cursor still in the parenthesis, use the "Select Columns" list box to locate and select the "InvoiceTotal" column.
11. Once selected, press the "Add Column" button to insert the column into the "Expression Text" panel.
12. Click the OK button to close the Expression Builder.
13. Click the OK button to close the column selection dialog and return to the Query Builder main window.

14. Right click on the "InvoiceHeader" table under "Tables/Views In Use", and choose "WHERE Clause" from the speed menu.
15. Use the "Select Columns" list box to locate and select the "TransDate" column.
16. Once selected, press the "Add Column" button to insert the column into the "Expression Text" panel.
17. From the "Operators and Keywords" panel, select the "BETWEEN" button to add the keyword to the WHERE Clause panel.
18. Type in the **third** quarter starting date of "07/01/2011".
19. From the "Operators and Keywords" panel, select the "AND" button to add the keyword to the WHERE Clause panel.
20. Type in the **third** quarter ending date of 09/30/2011".

The Expression panel should have the following displayed:

```
TransDate BETWEEN 07/01/2011 AND 09/30/2011
```

21. Click the OK button to close the WHERE Builder.

In the Query Builder main window, you should see the SQL syntax displayed in the bottom of the page as:

```
SELECT 1,(SUM( T1.InvoiceTotal ))
FROM  InvoiceHeader T1
WHERE TransDate BETWEEN 01/01/2011 AND 03/31/2011
UNION SELECT 2,(SUM( T1.InvoiceTotal ))
FROM  InvoiceHeader T1
WHERE T1.TransDate BETWEEN 04/01/2011 AND 06/30/2011
UNION SELECT 3,(SUM( T1.InvoiceTotal ))
FROM  InvoiceHeader T1
WHERE T1.TransDate BETWEEN 07/01/2011 AND 09/30/2011
```

1.1.4 4th Quarter

To add the 4th Quarter results, we will add a final UNION to the query.

1. Select the "Add UNION" button from Query Builder Toolbar.
2. Add the "InvoiceHeader" to the query by selecting the table, and then selecting the "Add Table/View" button.
3. Right click on the "InvoiceHeader" table under "Tables/Views In Use", and choose "Select Columns" from the speed menu.
4. In this example, select the "New Expression" button to display the Expression Builder.
5. Enter the value "4" into the "Expression Text" panel. The value "4" is the numeric value that will represent the 4th quarter of the figures in the query. Press the "OK" button to save the expression.
6. Select the "New Expression" button to once again display the Expression Builder.
7. Within the the "Function Templates" panel, select use the drop-down combo box to select the aggregate function "SUM", which will be displayed as "(SUM(arg))"
8. Select the "Use" button to add the function to the "Expression Text" panel.
9. Place the cursor within the set of parenthesis and delete the "arg" value.
10. Then, with the cursor still in the parenthesis, use the "Select Columns" list box to locate and select the "InvoiceTotal" column.
11. Once selected, press the "Add Column" button to insert the column into the "Expression Text" panel.
12. Click the OK button to close the Expression Builder.
13. Click the OK button to close the column selection dialog and return to the Query Builder main window.
14. Right click on the "InvoiceHeader" table under "Tables/Views In Use", and choose "WHERE Clause" from the speed menu.
15. Use the "Select Columns" list box to locate and select the "TransDate" column.
16. Once selected, press the "Add Column" button to insert the column into the "Expression Text" panel.
17. From the "Operators and Keywords" panel, select the "BETWEEN" button to add the keyword to the WHERE Clause panel.
18. Type in the **fourth** quarter starting date of "10/01/2011".
19. From the "Operators and Keywords" panel, select the "AND" button to add the keyword to the WHERE Clause panel.

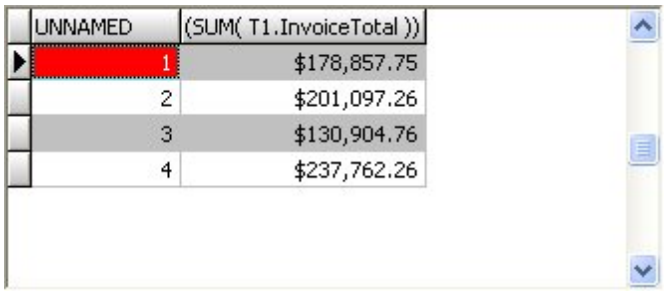
20.Type in the **fourth** quarter ending date of 12/31/2011".
21.Click the OK button to close the WHERE Builder.

In the Query Builder main window, you should see the SQL syntax displayed in the bottom of the page as:

```
SELECT 1,(SUM( T1.InvoiceTotal ))
FROM  InvoiceHeader T1
WHERE TransDate BETWEEN 01/01/2011 AND 03/31/2011
UNION SELECT 2,(SUM( T1.InvoiceTotal ))
FROM  InvoiceHeader T1
WHERE T1.TransDate  BETWEEN 04/01/2011 AND 06/30/2011
UNION SELECT 3,(SUM( T1.InvoiceTotal ))
FROM  InvoiceHeader T1
WHERE T1.TransDate BETWEEN 07/01/2011 AND 09/30/2011
UNION SELECT 4,(SUM( T1.InvoiceTotal ))
FROM  InvoiceHeader T1
WHERE T1.TransDate BETWEEN 10/01/2011 AND 12/31/2011
```

1.1.5 Browse the Query Results

At this point the query results can be reviewed by selecting the "Browse Query" button, which will display the data in the Data Browser.



UNNAMED	(SUM(T1.InvoiceTotal))
1	$178,857.75
2	$201,097.26
3	$130,904.76
4	$237,762.26

1.1.6 Column Aliases

With non-descriptive column headings, the Query Builder allow the ability to assign column aliases. By using column aliases, when the query is displayed, the results will be easier to understand.

1. If you have not done so already, close the Data Browser by pressing the [Esc] key.
2. From the main Menu Bar, choose "Query" > "Column Aliases.."

   The "Create Alias" dialog will appear.

3. First, enable aliases by selecting the "Use Column Aliases" check box.

   The panel will become editable with the "Column Name", "Alias Name", and "Modify" panels.

4. Select the first item in the "Column Name" panel.
5. The corresponding value displayed in the "Alias Name" will then appear in the "Modify" panel below.
6. Overwrite the value with the word "Quarter", and select the "Modify Current Selection" button.
7. Select the first item in the "Column Name" panel.
8. The corresponding value displayed in the "Alias Name" will then appear in the "Modify" panel below.
9. Overwrite the value with the word "TotalSales", and select the "Modify Current Selection" button.
10. Press the OK button to save the aliases.

In the Query Builder main window, you should see the SQL syntax displayed in the bottom of the page as:

```
(Quarter,TotalSales)
SELECT 1,(SUM( T1.InvoiceTotal ))
FROM  InvoiceHeader T1
WHERE TransDate BETWEEN 01/01/2011 AND 03/31/2011
UNION SELECT 2,(SUM( T1.InvoiceTotal ))
FROM  InvoiceHeader T1
WHERE T1.TransDate  BETWEEN 04/01/2011 AND 06/30/2011
UNION SELECT 3,(SUM( T1.InvoiceTotal ))
FROM  InvoiceHeader T1
WHERE T1.TransDate BETWEEN 07/01/2011 AND 09/30/2011
UNION SELECT 4,(SUM( T1.InvoiceTotal ))
FROM  InvoiceHeader T1
WHERE T1.TransDate BETWEEN 10/01/2011 AND 12/31/2011
```

1.1.7 Save the Query as a View

To display the alias values, first save the query.

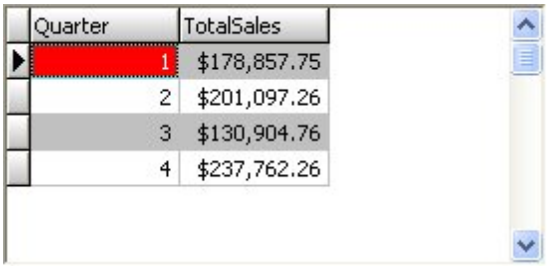
1. From the main Menu Bar, select "File" > "Save Query As View..."
2. In the dialog, enter "QuarterlySales2011" within the "View Name:" field.
3. In the dialog, enter "2011 Quarterly Sales Figures" within the "View Comment:" field.
4. Select the "OK" button.
5. Close the Query Builder by selecting "File" > "Close" from the Menu Bar.

In the Database Explorer, the view will be displayed with the name, comment, and number of columns

1.1.8 Browse the View

To launch the view, double click on "QuarterlySales2011"

The results should appear like the following.



Quarter	TotalSales
1	$178,857.75
2	$201,097.26
3	$130,904.76
4	$237,762.26

Part II

2 Building Queries Using Joins

A "Join" is an SQL clause that combines records from two tables in a database. When you perform a join, you specify one column from each table to join on. These two columns contain data that is shared across both tables. You can use multiple joins in the same SQL statement to query data from as many tables as you like. Join types include Inner, Left Outer, Right Outer, and Full Outer.

The differences are:

- **Inner Join** - returns rows when there is at least one row in both tables that match the join condition
- **Left Outer Join** - returns rows that have data in the left table (left of the JOIN keyword), even if there's no matching rows in the right table
- **Right Outer Join** - returns rows that have data in the right table (right of the JOIN keyword), even if there's no matching rows in the left table
- **Full Outer Join:** - returns all rows, as long as there's matching data in one of the tables

Most of the time, you'll do an Inner Join, though you will sometimes find it useful to do an Outer Join. For example, you need an Outer Join to get all rows in the following cases:

- When joining a *customer* table with an *orders* table to list the customers who ordered something in the current month as well as those who didn't order anything.
- When joining a *budget* table with an *expense* table to list each budget item, whether or not there was an expense for that item in the current month.
- When comparing a header (master table) on the "one" side of a one-to-many relationship against a detail (transaction table) on the "many" side to see all the rows of data, whether or not they have associated details.

2.1 Inner Join

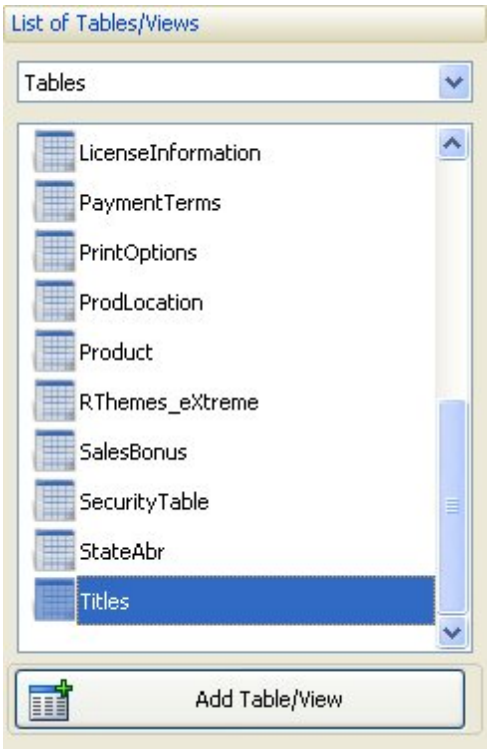
When using an Inner Join, rows are returned when there is at least one row in both tables that match the join condition. Inner Joins are the most common join operation and is the default for R:BASE joins.

The following instructions will step through the process to recreate an Inner Join in the Query Builder.

To continue, launch R:BASE eXtreme 9.5 and connect to the RRBYW18 sample database. If you are using R:BASE eXtreme 9.5 (64), the database is located in the following default installation directory: "C:\RBTI\RBG95_64\Samples\RRBYW18". If you are using R:BASE eXtreme 9.5 (32), the database is located in the following default installation directory: "C:\RBTI\RBG95_32\Samples\RRBYW18".

1. Start R:BASE eXtreme 9.5
2. Connect to the RRBYW18 sample database, by selecting "Database" > "Connect", and navigating to the above default installation directory based upon your version.
3. Then, launch the Query Builder by selecting "Tools" > "Query by Example" from the main Menu Bar

   In the Query Builder you will see a list of tables and views within a panel to the left that can be added to your query. The list contains all tables and views for the connected database.

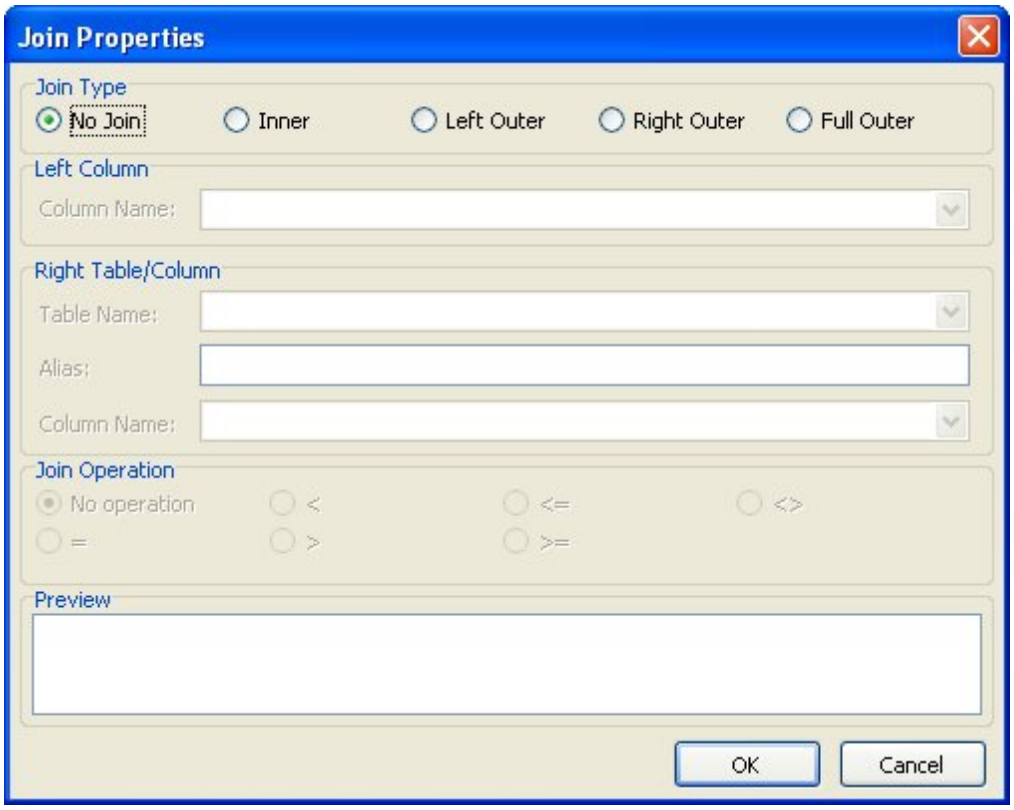


4. Add the "Titles" table to the query by selecting the table, and then selecting the "Add Table/View" button.

   The table should now be listed under "Tables/Views In Use".

5. Right click on the "Titles" table under "Tables/Views In Use", and choose "Join Properties..." from the speed menu.

   The "Join Properties" dialog will be displayed.

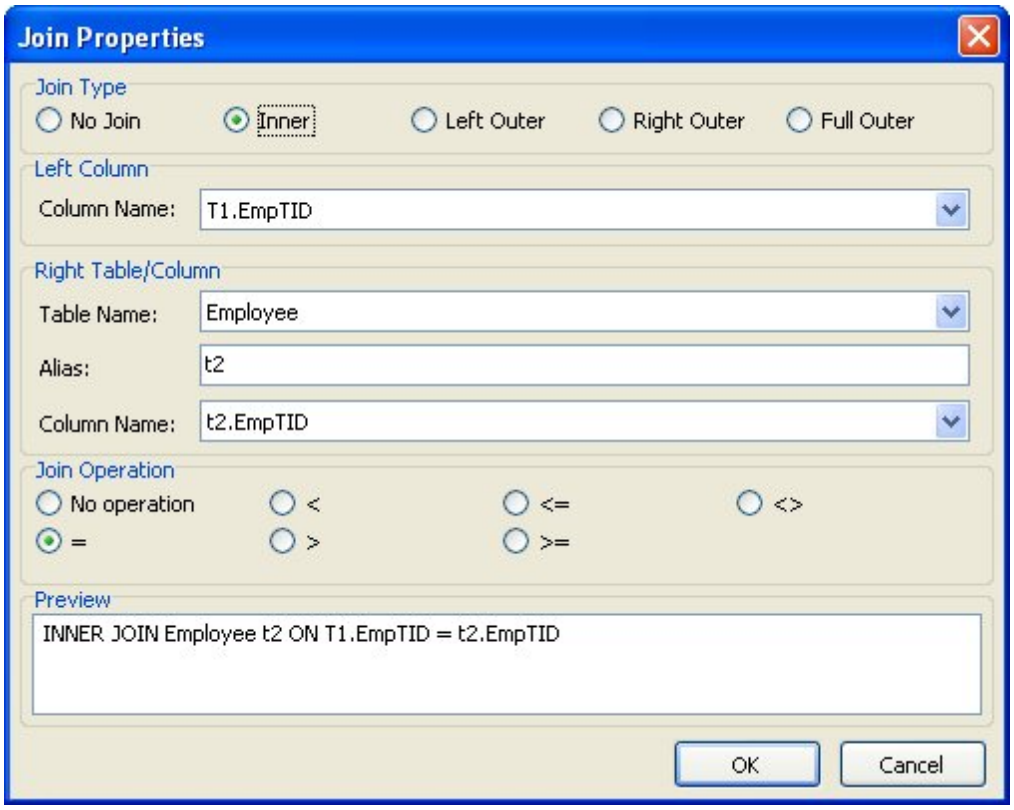


6. From the "Join Type" radio button options, select "Inner"

   Take note that the other options within the window will become enabled. In this window you will select the linking columns for the join, the second table for the join, and the join operation. As alias can also be assigned to the linking column.

7. From the "Left Column" panel, choose "T1.EmpTID" from the "Column Name:" drop down box. The "T1" alias may vary from your R:BASE screen.
8. From the "Right Table/Column" panel, choose "Employee" from the "Table Name:" drop down box.
9. From the "Right Table/Column" panel, enter "T2" into the "Alias:" field to assign the table alias for the join. If "T2" is already used for the "Titles" table, then use "T3" as the alias.
10. Again in the "Right Table/Column" panel, choose the "EmpTID" column from the "Column Name:" drop down box. The alias you have assigned will appear in from of the column name.
11. From the "Join Operation" radio button options, choose the equal character (=).

    You end result for the Join Properties should look like, or close to, the following:

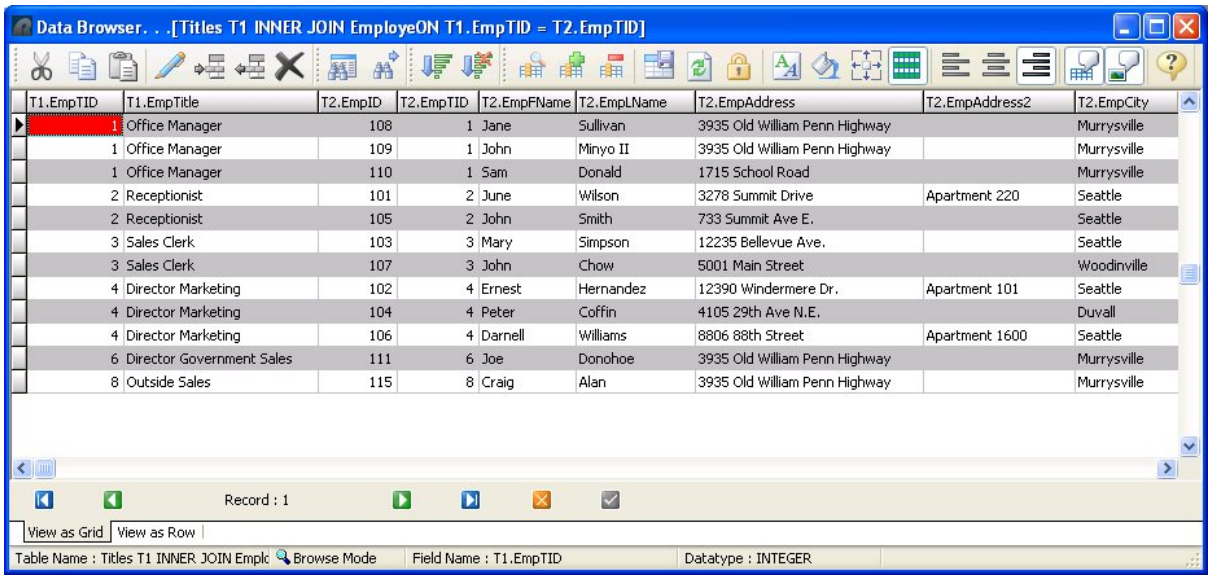


12. Select the "OK" button.

    In the Query Builder main window, you should see the SQL syntax displayed in the bottom of the page as:

```
SELECT *
FROM  Titles T1 INNER JOIN Employee t2 ON T1.EmpTID = t2.EmpTID
```

13. Then, browse the query results by selecting "Query" > "Browse Query" from the main Menu Bar. The results should look like the following:



T1.EmpTID	T1.EmpTitle	T2.EmpID	T2.EmpTID	T2.EmpFName	T2.EmpLName	T2.EmpAddress	T2.EmpAddress2	T2.EmpCity
1	Office Manager	108	1	Jane	Sullivan	3935 Old William Penn Highway		Murrysville
1	Office Manager	109	1	John	Minyo II	3935 Old William Penn Highway		Murrysville
1	Office Manager	110	1	Sam	Donald	1715 School Road		Murrysville
2	Receptionist	101	2	June	Wilson	3278 Summit Drive	Apartment 220	Seattle
2	Receptionist	105	2	John	Smith	733 Summit Ave E.		Seattle
3	Sales Clerk	103	3	Mary	Simpson	12235 Bellevue Ave.		Seattle
3	Sales Clerk	107	3	John	Chow	5001 Main Street		Woodinville
4	Director Marketing	102	4	Ernest	Hernandez	12390 Windermere Dr.	Apartment 101	Seattle
4	Director Marketing	104	4	Peter	Coffin	4105 29th Ave N.E.		Duvall
4	Director Marketing	106	4	Darnell	Williams	8806 88th Street	Apartment 1600	Seattle
6	Director Government Sales	111	6	Joe	Donohoe	3935 Old William Penn Highway		Murrysville
8	Outside Sales	115	8	Craig	Alan	3935 Old William Penn Highway		Murrysville

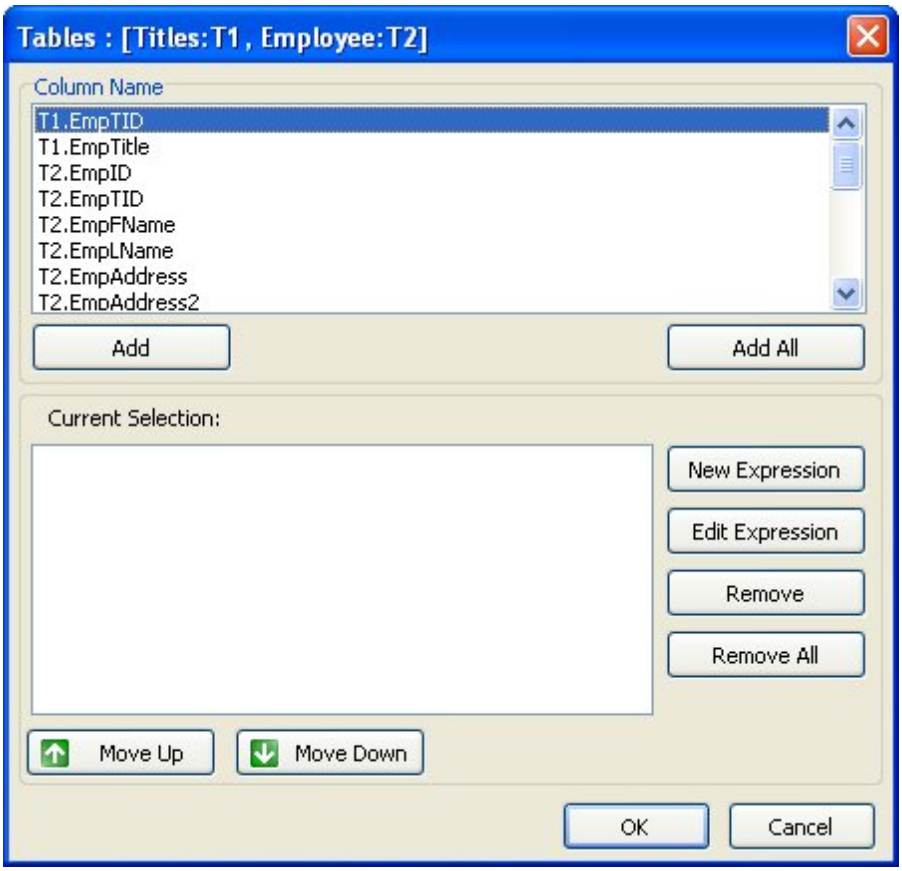
The results show the "Employee" table added to the "Titles" table. Each employee is displayed next to the title where the "EmpID" column matches.

14. Close the Data Browser window.

    Next, the number of columns displayed will be decreased.

15. Right click on the "Titles" table under "Tables/Views In Use", and choose "Select Columns..." from the speed menu.

    A dialog will be displayed with the available columns for each of the tables.



Once the "Tables" dialog is displayed, you can add the columns one at a time by selecting the desired column and pressing the "Add" button, or double click on a desired column. All columns can be added at once by pressing the "Add All" button.

16. Select the "EmpTitle" column, and select the "Add" button.

When a column is added, an optional column alias can be assigned in the "Column Alias" dialog, that appears each time a column is added to the query/view.

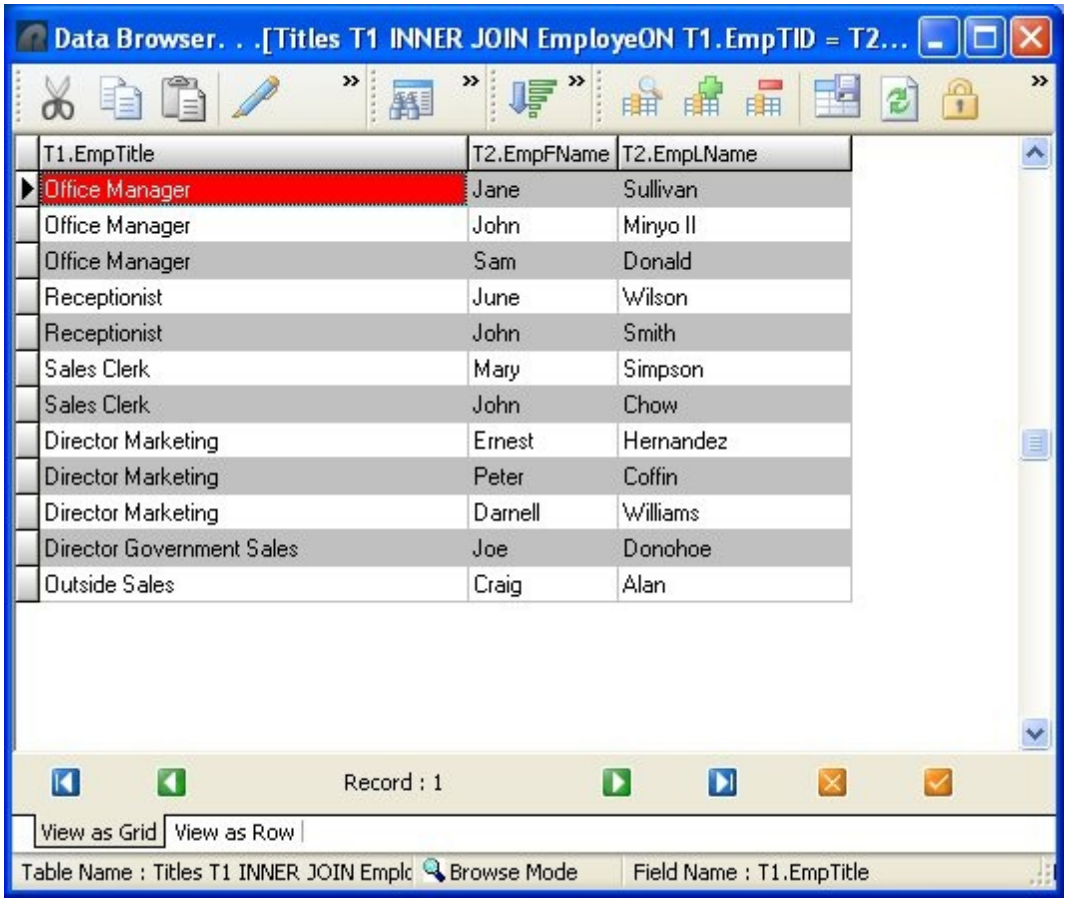
17. Press the "OK" or "Cancel" buttons to allow R:BASE to assign the alias for you.
18. Also add the "EmpFName" and "EmpLName" columns individually and do not enter an alias for each.
19. Press the "OK" button to save the columns added.

In the Query Builder main window, you should see the SQL syntax displayed in the bottom of the page as:

```
SELECT T1.EmpTitle,T2.EmpFName,T2.EmpLName
```

```
FROM  Titles T1 INNER JOIN Employee T2 ON T1.EmpTID = T2.EmpTID
```

20. Now, browse the query results by selecting "Query" > "Browse Query" from the main Menu Bar. The results should look like the following:



The results show the "Employee" table added to the "Titles" table and only the title first name and last name columns are displayed.Each employee is displayed next to the title where the "EmpID" column matches.

If you wish to save the query, follow the below steps.

1. From the main Menu Bar, select "File" > "Save Query As View..."
2. In the dialog, enter "TitleEmp" within the "View Name:" field.
3. In the dialog, enter "Title and Employee List" within the "View Comment:" field.
4. Select the "OK" button.
5. Close the Query Builder by selecting "File" > "Close" from the Menu Bar.

In the Database Explorer, the view will be displayed with the name, comment, and number of columns.

2.2 Left Outer Join

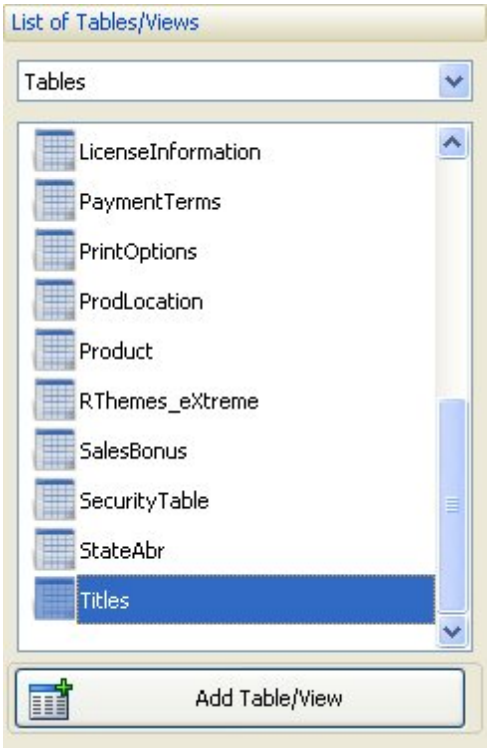
When using an Outer Join, rows are not required to have matching values. The table order in the FROM clause specifies the left and right table. For a Left Outer Join, R:BASE uses each value unique to the left (first) table and completes it with nulls for the columns of the right (second) table when the linking columns do not match.

The following instructions will step through the process to recreate a Left Outer Join in the Query Builder.

To continue, launch R:BASE eXtreme 9.5 and connect to the RRBYW18 sample database. If you are using R:BASE eXtreme 9.5 (64), the database is located in the following default installation directory: "C:\RBTI\RBG95_64\Samples\RRBYW18". If you are using R:BASE eXtreme 9.5 (32), the database is located in the following default installation directory: "C:\RBTI\RBG95_32\Samples\RRBYW18".

1. Start R:BASE eXtreme 9.5
2. Connect to the RRBYW18 sample database, by selecting "Database" > "Connect", and navigating to the above default installation directory based upon your version.
3. Then, launch the Query Builder by selecting "Tools" > "Query by Example" from the main Menu Bar

   In the Query Builder you will see a list of tables and views within a panel to the left that can be added to your query. The list contains all tables and views for the connected database.

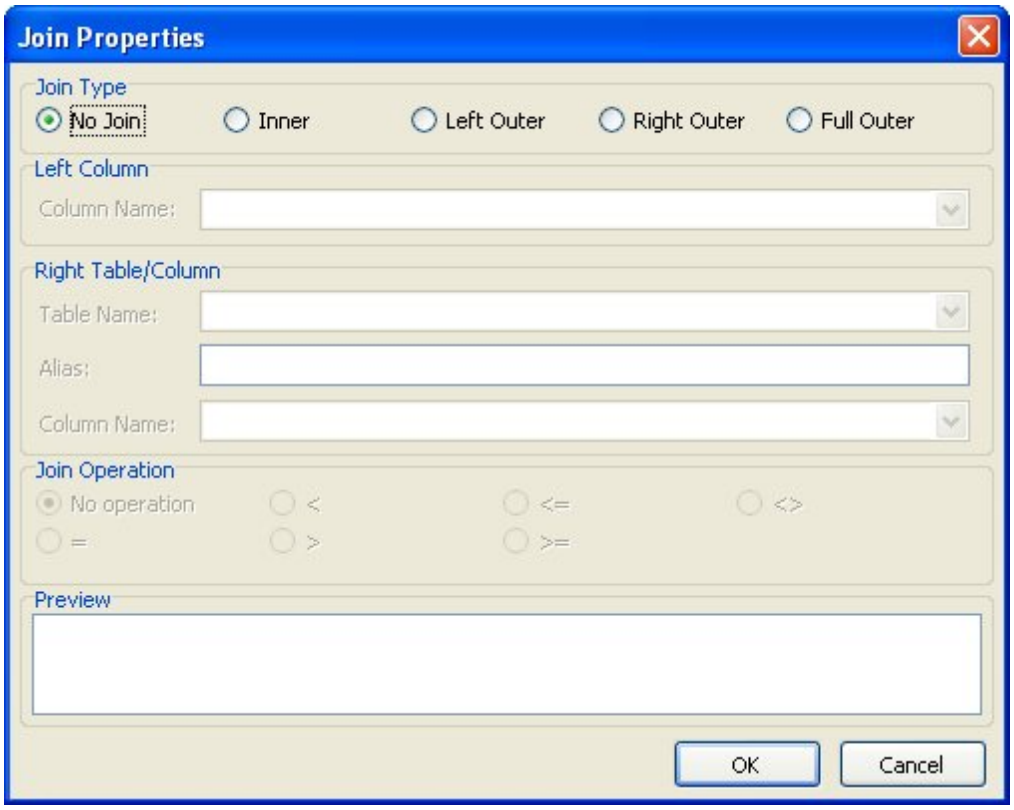


4. Add the "Titles" table to the query by selecting the table, and then selecting the "Add Table/View" button.

   The table should now be listed under "Tables/Views In Use".

5. Right click on the "Titles" table under "Tables/Views In Use", and choose "Join Properties..." from the speed menu.

   The "Join Properties" dialog will be displayed.

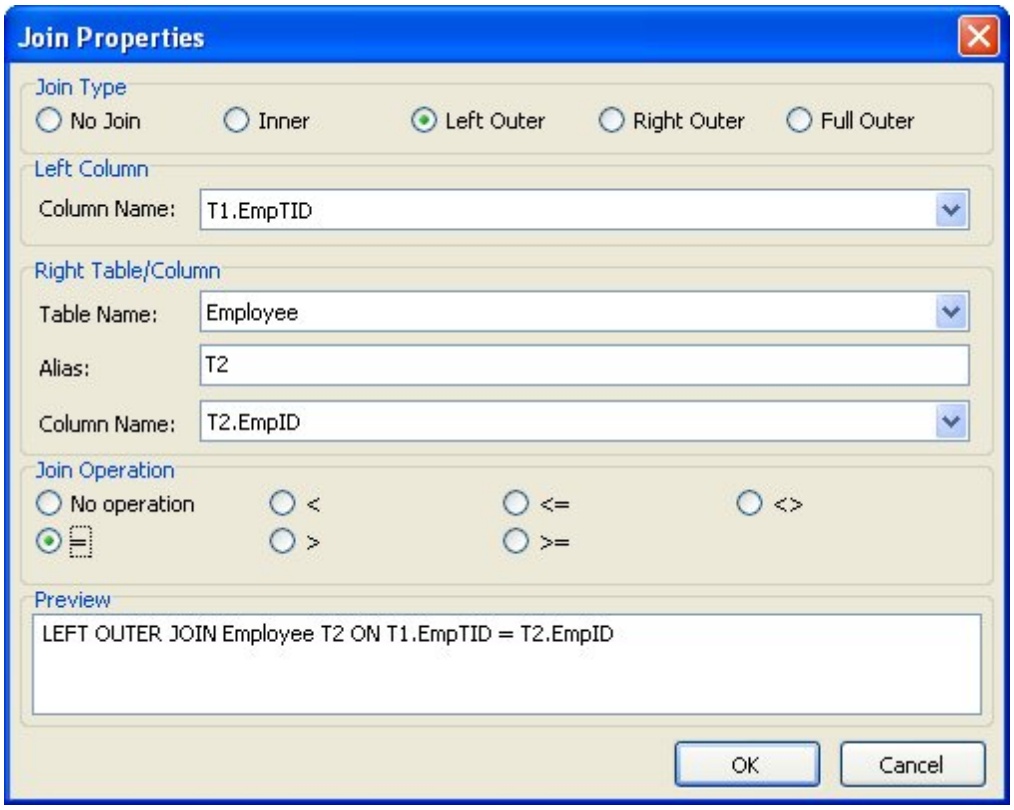


6. From the "Join Type" radio button options, select "Left Outer"

   Take note that the other options within the window will become enabled. In this window you will select the linking columns for the join, the second table for the join, and the join operation. As alias can also be assigned to the linking column.

7. From the "Left Column" panel, choose "T1.EmpTID" from the "Column Name:" drop down box. The "T1" alias may vary from your R:BASE screen.
8. From the "Right Table/Column" panel, choose "Employee" from the "Table Name:" drop down box.
9. From the "Right Table/Column" panel, enter "T2" into the "Alias:" field to assign the table alias for the join. If "T2" is already used for the "Titles" table, then use "T3" as the alias.
10. Again in the "Right Table/Column" panel, choose the "EmpTID" column from the "Column Name:" drop down box. The alias you have assigned will appear in from of the column name.
11. From the "Join Operation" radio button options, choose the equal character (=).

    You end result for the Join Properties should look like, or close to, the following:



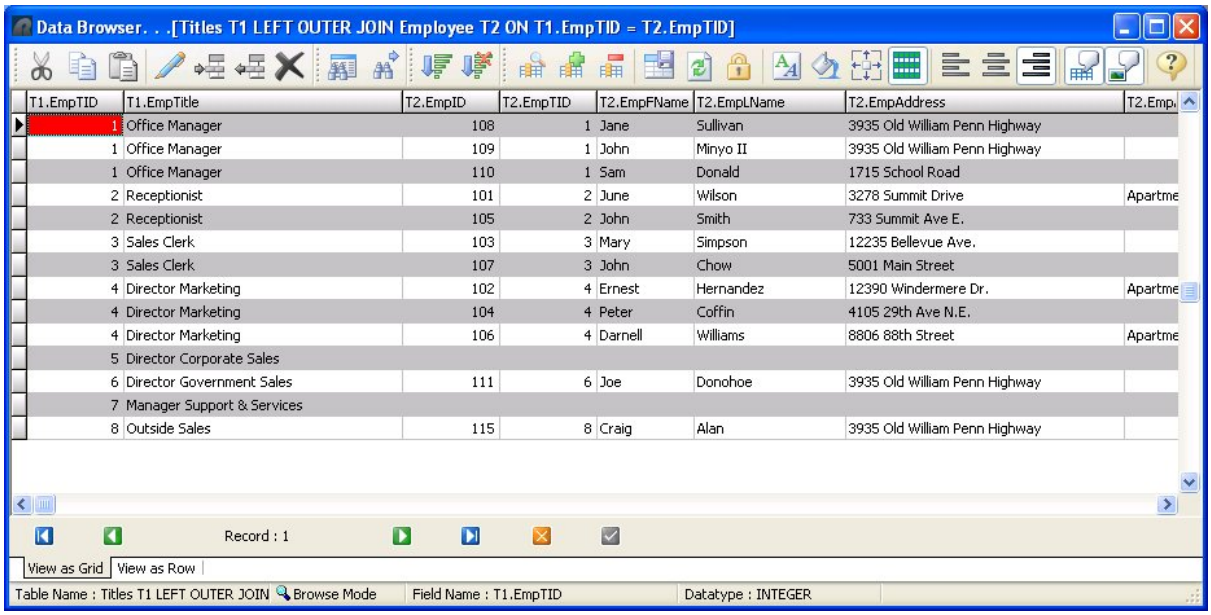
12. Select the "OK" button.

    In the Query Builder main window, you should see the SQL syntax displayed in the bottom of the page as:

    ```
    SELECT *
    FROM  Titles T1 LEFT OUTER JOIN Employee T2 ON T1.EmpTID = T2.EmpID
    ```

    The asterisk (*) in the command syntax represents all columns. The next steps will show how to specify certain columns to display.

13. Now, browse the query results by selecting "Query" > "Browse Query" from the main Menu Bar. The results should look like the following:



The results show the Employee table added to the Titles table. Where no employees match for an existing title, the row is empty, or null. In the above, there are no employees that have the "Director Corporate Sales" or "Manager Support & Services" titles.

If you wish to save the query, follow the below steps.

1. From the main Menu Bar, select "File" > "Save Query As View..."
2. In the dialog, enter "TitleEmpList" within the "View Name:" field.
3. In the dialog, enter "Title List with Employees" within the "View Comment:" field.
4. Select the "OK" button.
5. Close the Query Builder by selecting "File" > "Close" from the Menu Bar.

In the Database Explorer, the view will be displayed with the name, comment, and number of columns.

2.3 Full Outer Join

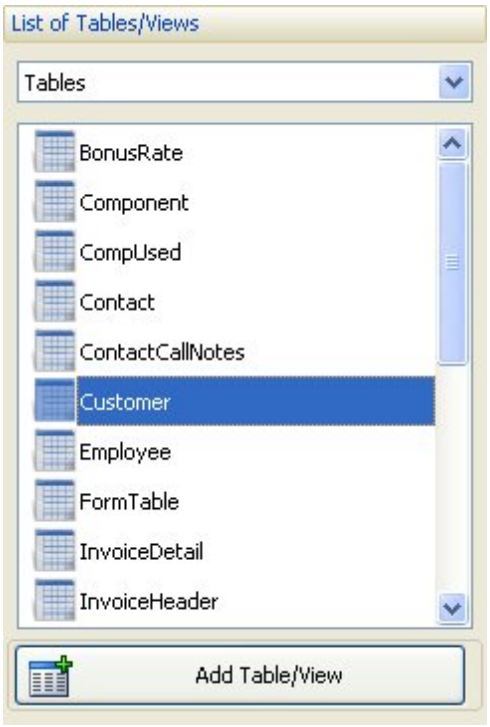
When using an Outer Join, rows are not required to have matching values. The table order in the FROM clause specifies the left and right table. For a Full Outer Join, R:BASE returns all rows, as long as there's matching data in one of the tables.

The following instructions will step through the process to recreate a Full Outer Join in the Query Builder.

To continue, launch R:BASE eXtreme 9.5 and connect to the RRBYW18 sample database. If you are using R:BASE eXtreme 9.5 (64), the database is located in the following default installation directory: "C:\RBTI\RBG95_64\Samples\RRBYW18". If you are using R:BASE eXtreme 9.5 (32), the database is located in the following default installation directory: "C:\RBTI\RBG95_32\Samples\RRBYW18".

1. Start R:BASE eXtreme 9.5
2. Connect to the RRBYW18 sample database, by selecting "Database" > "Connect", and navigating to the above default installation directory based upon your version.
3. Then, launch the Query Builder by selecting "Tools" > "Query by Example" from the main Menu Bar

In the Query Builder you will see a list of tables and views within a panel to the left that can be added to your query. The list contains all tables and views for the connected database.

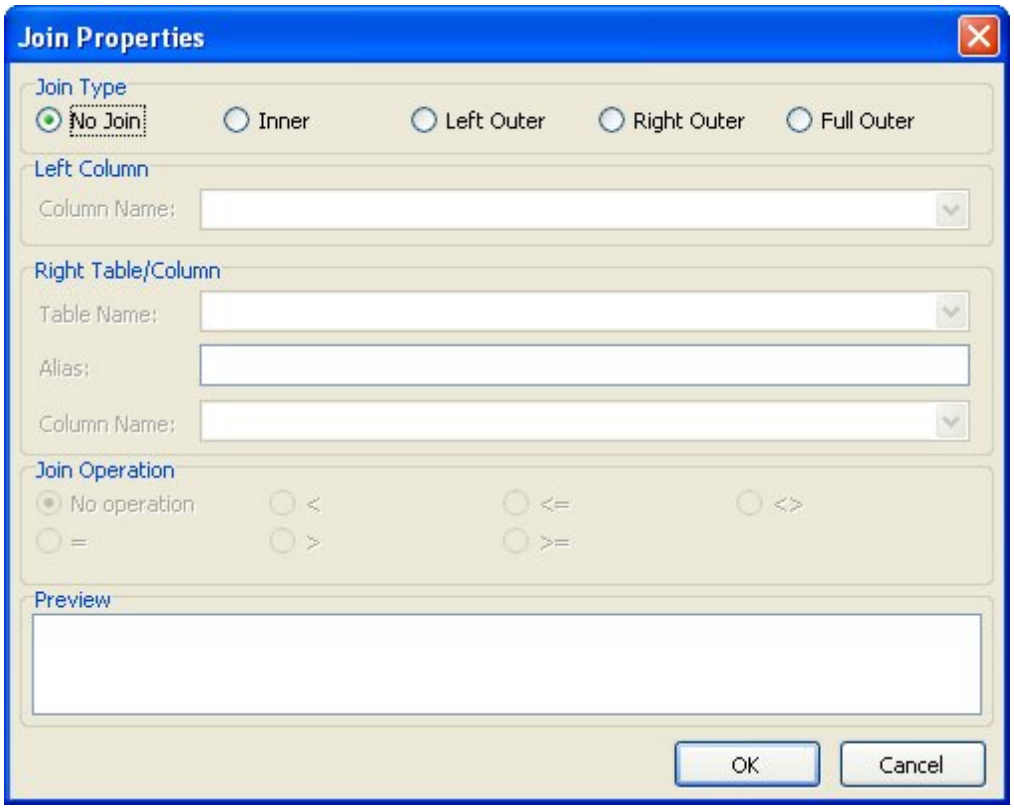


4. Add the "Customer" table to the query by selecting the table, and then selecting the "Add Table/View" button.

   The table should now be listed under "Tables/Views In Use".

5. Right click on the "Customer" table under "Tables/Views In Use", and choose "Join Properties..." from the speed menu.

   The "Join Properties" dialog will be displayed.

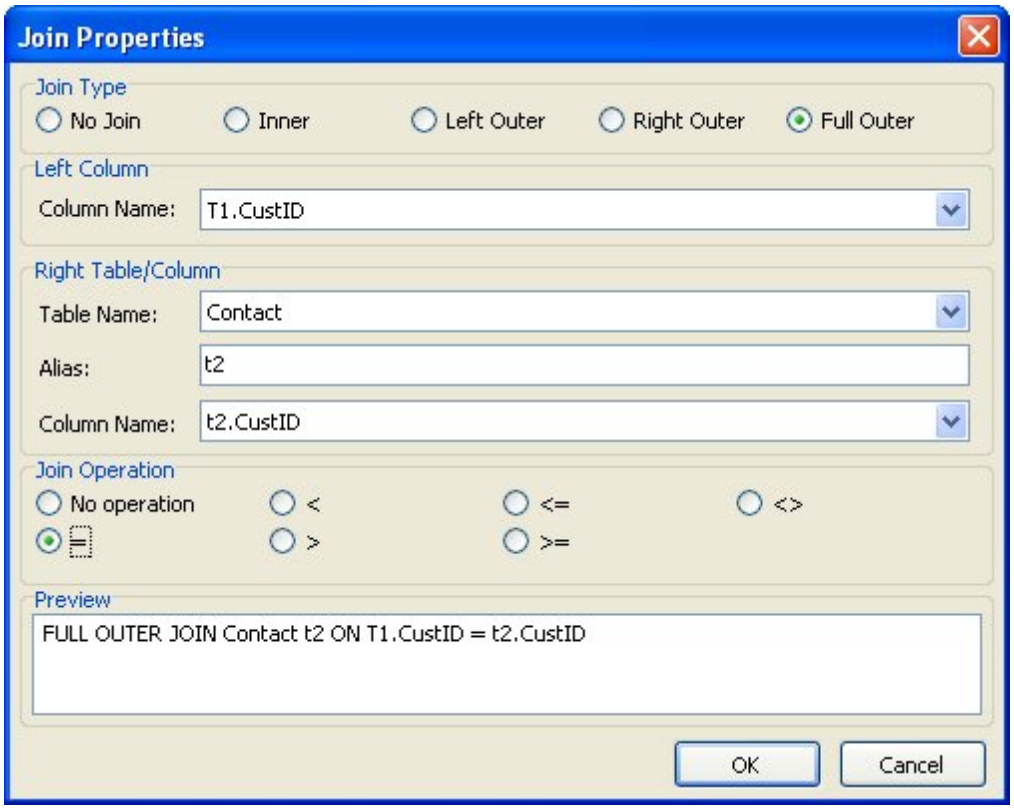


6. From the "Join Type" radio button options, select "Full Outer"

   Take note that the other options within the window will become enabled. In this window you will select the linking columns for the join, the second table for the join, and the join operation. As alias can also be assigned to the linking column.

7. From the "Left Column" panel, choose "T1.CustID" from the "Column Name:" drop down box. The "T1" alias may vary from your R:BASE screen.
8. From the "Right Table/Column" panel, choose "Contact" from the "Table Name:" drop down box.
9. From the "Right Table/Column" panel, enter "T2" into the "Alias:" field to assign the table alias for the join. If "T2" is already used for the "Customer" table, then use "T3" as the alias.
10. Again in the "Right Table/Column" panel, choose the "CustID" column from the "Column Name:" drop down box. The alias you have assigned will appear in from of the column name.
11. From the "Join Operation" radio button options, choose the equal character (=).

    You end result for the Join Properties should look like, or close to, the following:



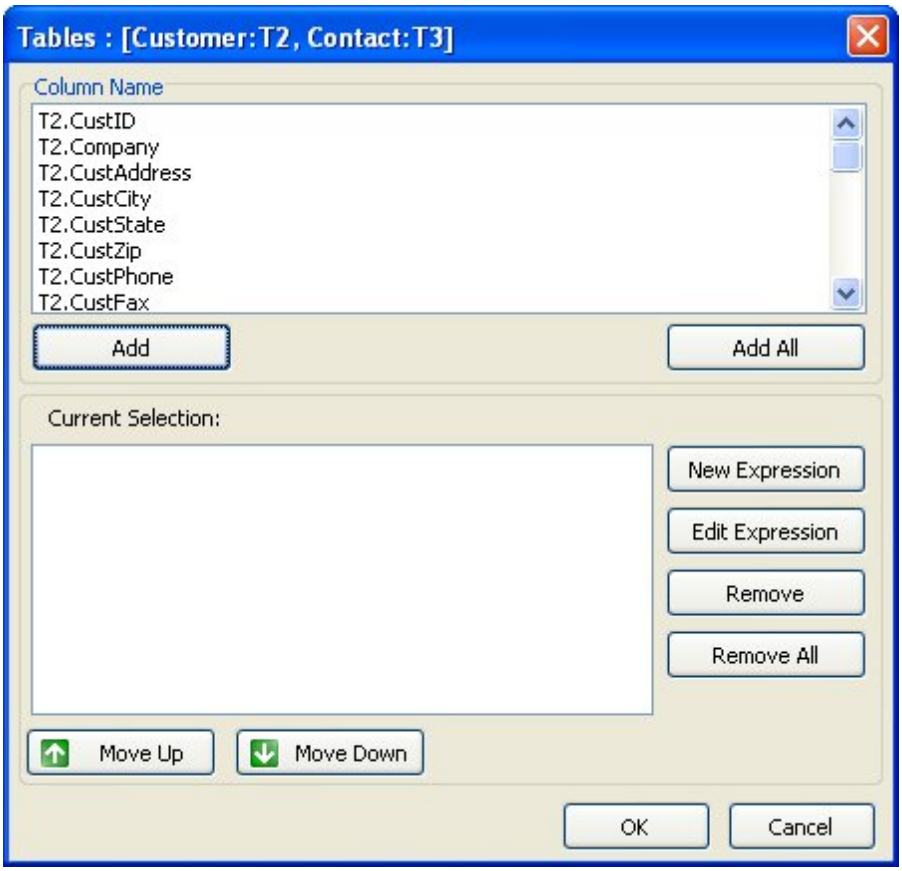
12. Select the "OK" button.

    In the Query Builder main window, you should see the SQL syntax displayed in the bottom of the page as:

```
SELECT *
FROM  Customer T1 FULL OUTER JOIN Contact t2 ON T1.CustID = t2.CustID
```

13. Right click on the "Customer" table under "Tables/Views In Use", and choose "Select Columns..." from the speed menu.

    A dialog will be displayed with the available columns for each of the tables.



Once the "Tables" dialog is displayed, you can add the columns one at a time by selecting the desired column and pressing the "Add" button, or double click on a desired column. All columns can be added at once by pressing the "Add All" button.

14. Select the "Company" column, and select the "Add" button.

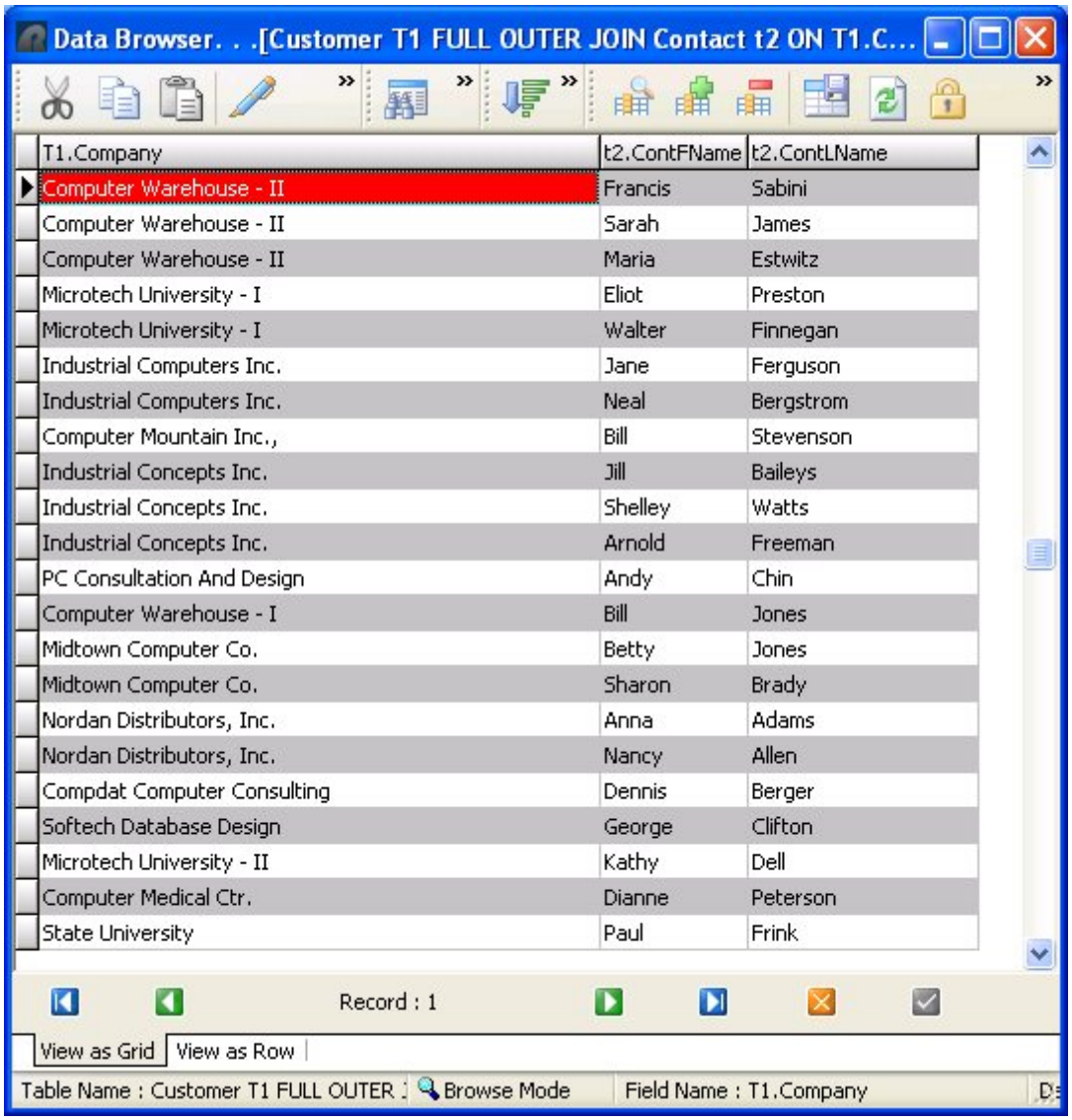
When a column is added, an optional column alias can be assigned in the "Column Alias" dialog, that appears each time a column is added to the query/view.

15. Press the "OK" or "Cancel" buttons to allow R:BASE to assign the alias for you.
16. From the "Contacts" table, add the "ContFName" and "ContLName" columns individually, and do not enter an alias for each.
17. Press the "OK" button to save the columns added.

In the Query Builder main window, you should see the SQL syntax displayed in the bottom of the page as:

```
SELECT T2.Company,T3.ContFName,T3.ContLName
FROM  Customer T2 FULL OUTER JOIN Contact T3 ON T2.CustID = T3.CustID
```

18. Now, browse the query results by selecting "Query" > "Browse Query" from the main Menu Bar. The results should look like the following:



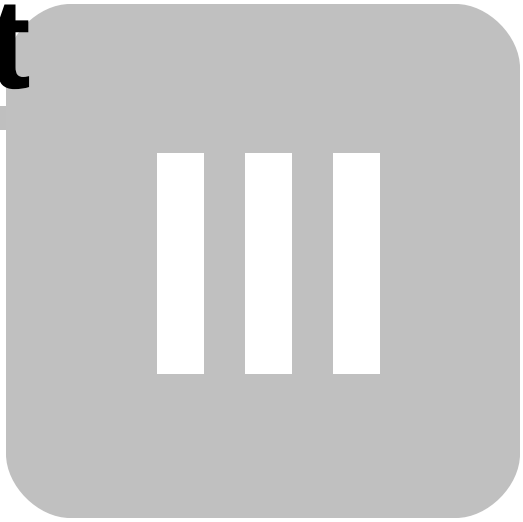
The results show the Contact table added to the Customer table. Notice that where any Contact matches for an existing Customer, the company is listed. For example, since there are three contacts for the "Computer Warehouse - II" the company name is listed for each contact.

If you wish to save the query, follow the below steps.

1. From the main Menu Bar, select "File" > "Save Query As View..."
2. In the dialog, enter "CustContacts" within the "View Name:" field.
3. In the dialog, enter "Customer and Contact List" within the "View Comment:" field.
4. Select the "OK" button.
5. Close the Query Builder by selecting "File" > "Close" from the Menu Bar.

In the Database Explorer, the view will be displayed with the name, comment, and number of columns.

Part



3 Building Queries Using GROUP BY

A GROUP BY clause groups rows according to the values in one or more columns and sorts the results. GROUP BY consolidates the information from several rows into one row. This results in a table with one row for each value in the named column or columns and one or more values per column. With the GROUP BY the columns can be sorted in ascending or descending order.

The columns listed in the GROUP BY clause are related to those listed in the query's selected columns. Any column named in the GROUP BY clause can also be named in the query's selected columns, but any column not named in the GROUP BY clause can be used only in the selected columns if the column is used in a SELECT function (SUM, COUNT, etc.).

An optional HAVING clause can also be used with a GROUP BY.

HAVING Clause
The HAVING clause determines which rows of data to include based on the results of the GROUP BY clause, and essentially limits the rows affected by the GROUP BY clause.

The HAVING clause selects rows that meet one or more conditions from among the results of the GROUP BY clause. HAVING works the same as a WHERE clause with the following exceptions:

- A WHERE clause modifies the intermediate results of a FROM clause; a HAVING clause modifies the intermediate results of a GROUP BY clause
- A HAVING clause can include SELECT Functions

3.1 Grouping Employee Job Titles

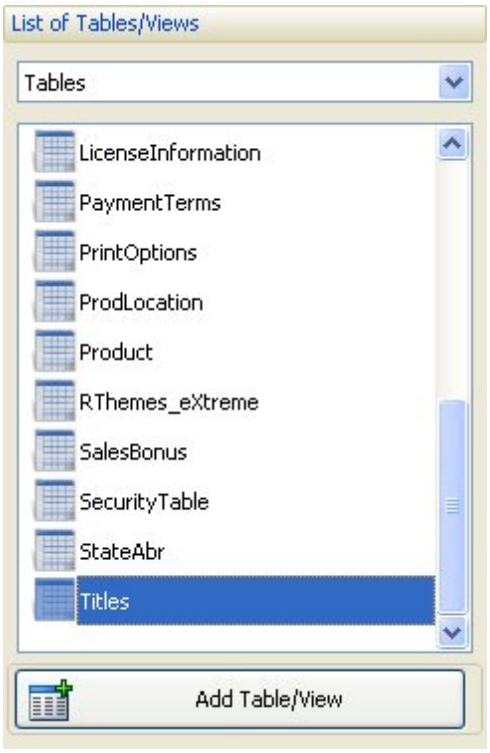
A GROUP BY can be used to consolidate information from several rows into one row.

The following instructions will step through the process to recreate a view using a GROUP BY with the COUNT Function in the Query Builder.

To continue, launch R:BASE eXtreme 9.5 and connect to the RRBYW18 sample database. If you are using R:BASE eXtreme 9.5 (64), the database is located in the following default installation directory: "C:\RBTI\RBG95_64\Samples\RRBYW18". If you are using R:BASE eXtreme 9.5 (32), the database is located in the following default installation directory: "C:\RBTI\RBG95_32\Samples\RRBYW18".

1. Start R:BASE eXtreme 9.5
2. Connect to the RRBYW18 sample database, by selecting "Database" > "Connect", and navigating to the above default installation directory based upon your version.
3. Then, launch the Query Builder by selecting "Tools" > "Query by Example" from the main Menu Bar

   In the Query Builder you will see a list of tables and views within a panel to the left that can be added to your query. The list contains all tables and views for the connected database.



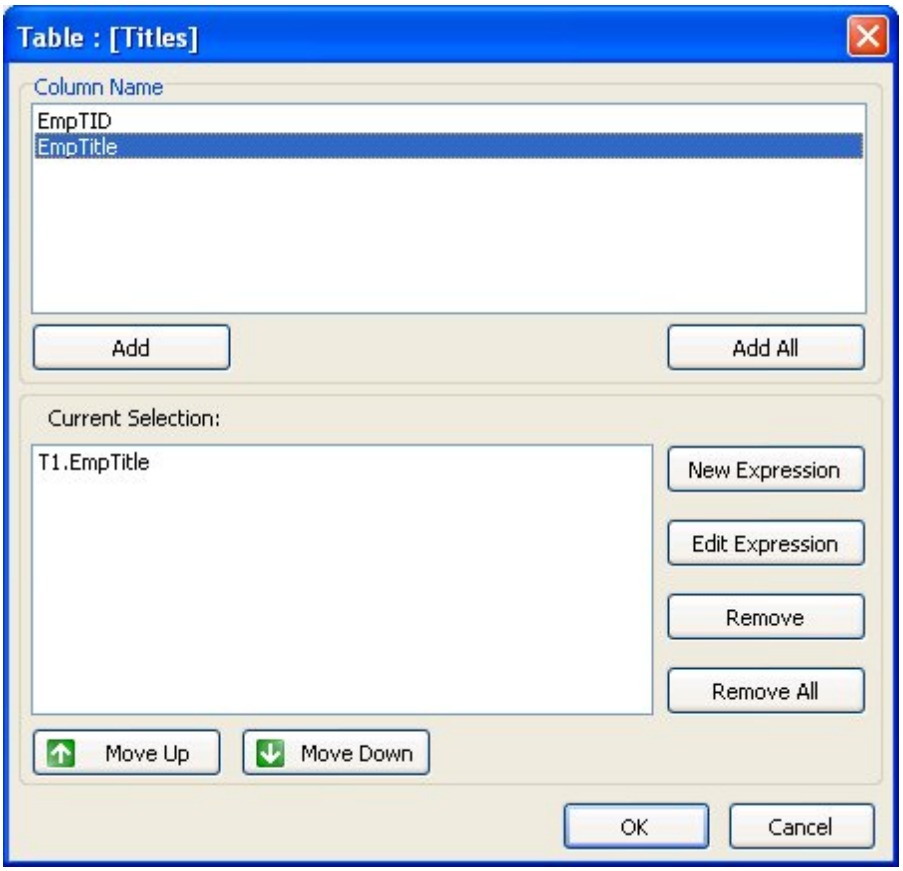
4. Add the "Titles" table to the query by selecting the table, and then selecting the "Add Table/View" button.

   The table should now be listed under "Tables/Views In Use".

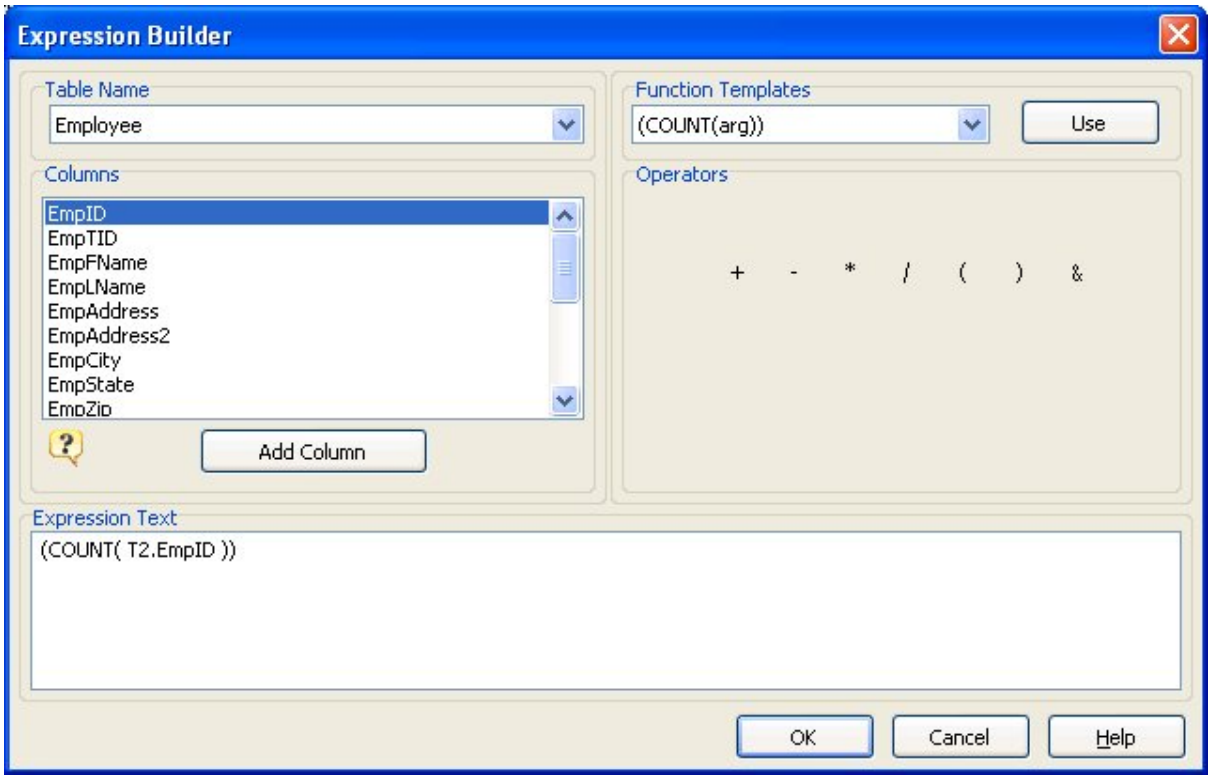
5. Add the "Employee" table to the query by selecting the table, and then selecting the "Add Table/View" button.

   The table should now be listed under "Tables/Views In Use". With the tables added, the specific column(s) to be used in the query can be selected.

6. Right click on the "Titles" table and select "Select Columns ..."

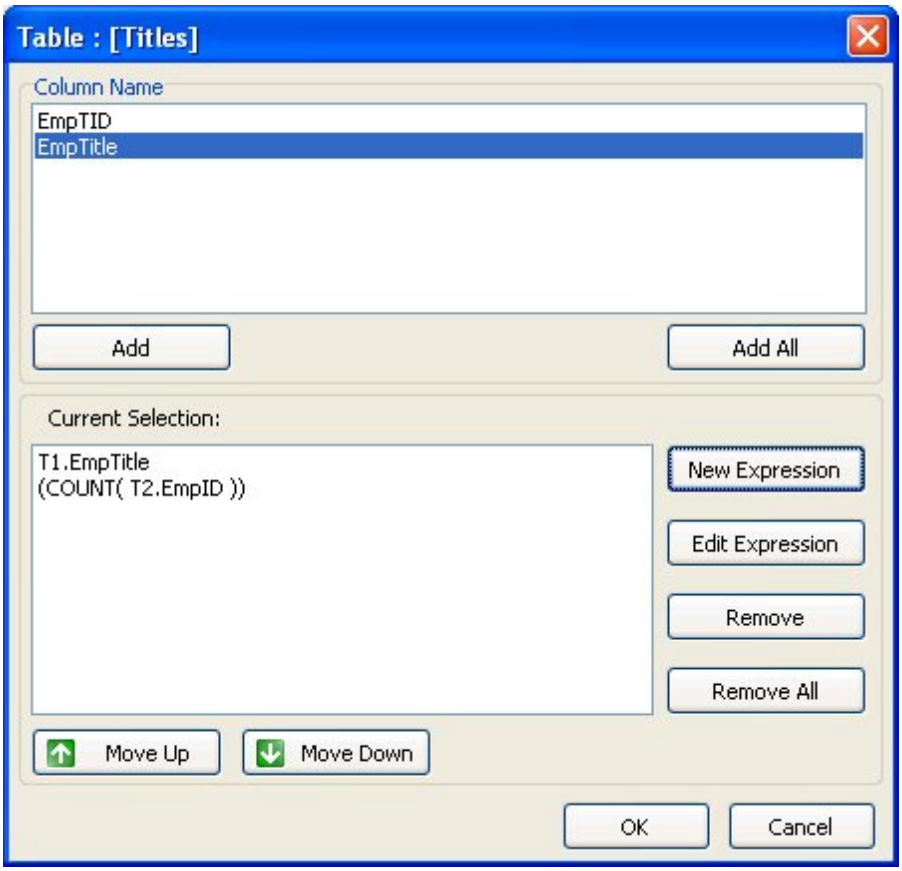


7. Add the "EmpTitle" column. When prompted for a column alias, just select the OK button.
8. Click on the "New Expression" button.
9. Using the "Table Name" drop down box, select "Employee" as the table name.
10. From the "Function Templates", select "(COUNT(arg))", then the "Use" button. The COUNT function will appear in the "Expression Text" panel.

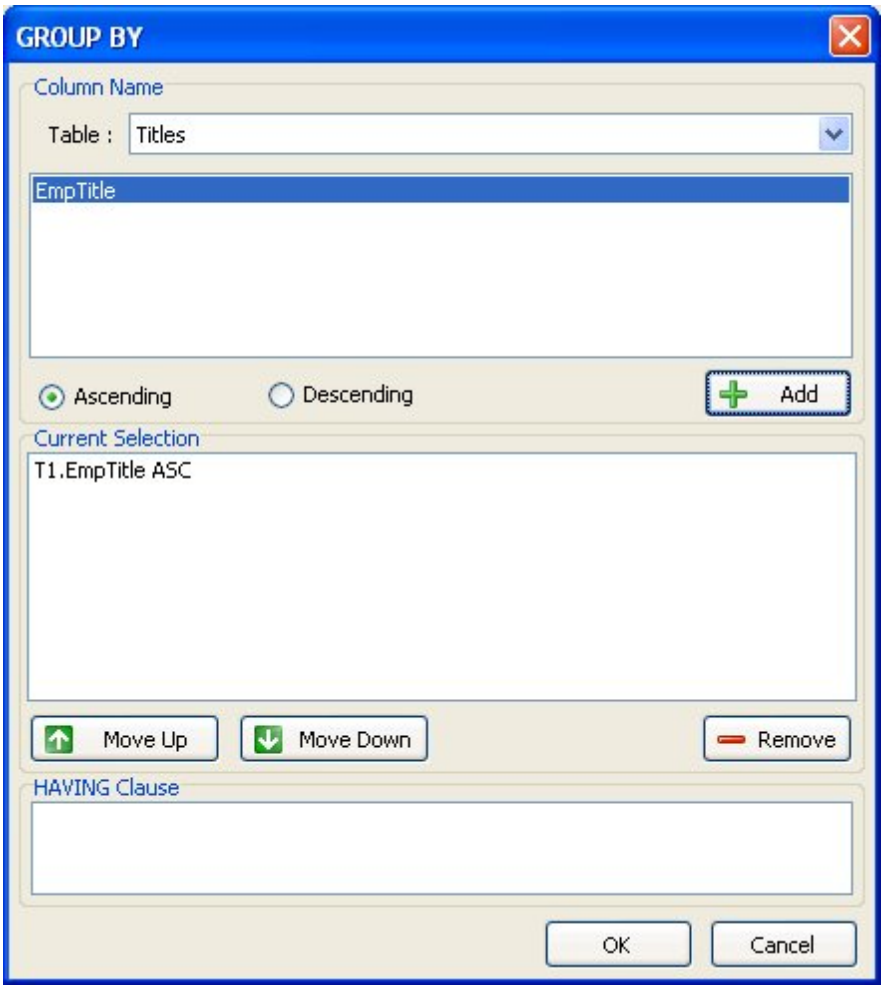


11. Within the "Expression Text" panel, remove the text "arg", and select "EmpID" from the list of "Columns", then select the "Add Column" button.
12. Select the "OK" button.

The "Current Selection" results should look like this:



13. Select the "OK" button to return to the Query Builder.
14. Right click on the "Titles" table under "Tables/Views In Use", and choose "GROUP BY" from the speed menu.

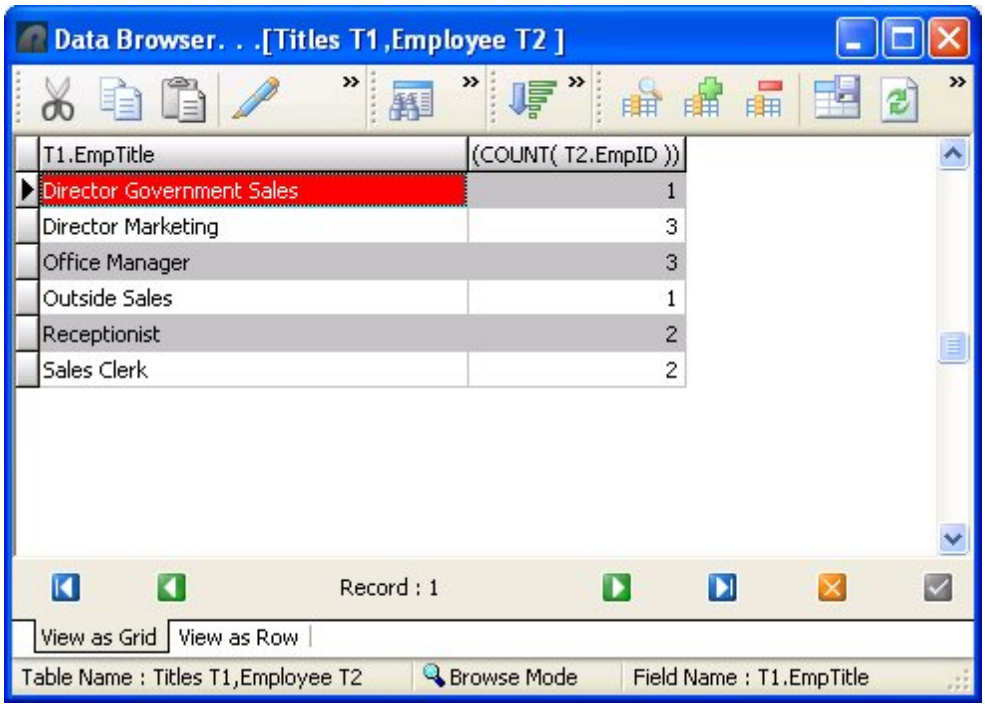


15. Select the "EmpTitle" column, leave the "Ascending" radio button selected, and click the "Add" button.
16. Select the "OK" button to return to the Query Builder.

    In the Query Builder main window, you should see the SQL syntax displayed in the bottom of the page as:

```
SELECT T1.EmpTitle,(COUNT( T2.EmpID ))
FROM  Titles T1, Employee T2
WHERE T1.EmpTID = T2.EmpTID
GROUP BY T1.EmpTitle
```

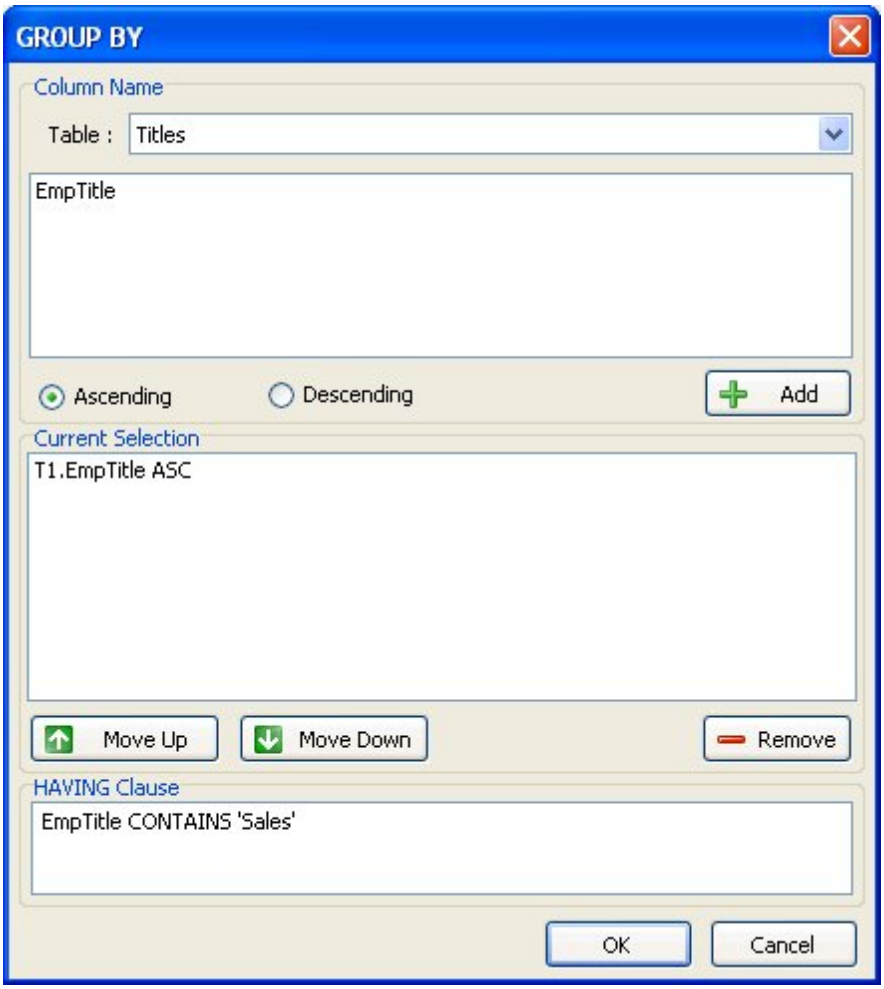
17. Now, browse the query results by selecting "Query" > "Browse Query" from the main Menu Bar. The results should look like the following:



The results should display a column of employee titles, and the number of employees with each title. Close the Data Browser window.

A HAVING clause can be added to only display job titles with "sales" in the name.

18. Right click on the "Titles" table under "Tables/Views In Use", and choose "GROUP BY" from the speed menu.

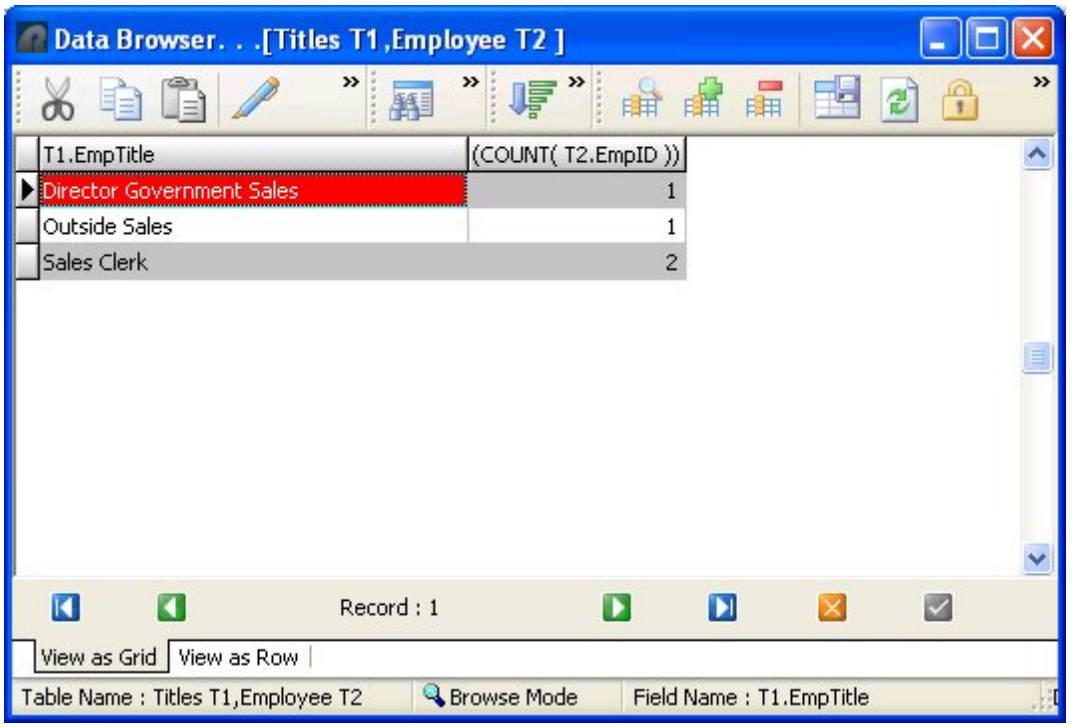


19. Within the "HAVING Clause" panel, enter: EmpTitle CONTAINS 'Sales'
20. Select the "OK" button to return to the Query Builder.

    In the Query Builder main window, you should see the SQL syntax displayed in the bottom of the page as:

```
SELECT T1.EmpTitle,(COUNT( T2.EmpID ))
FROM  Titles T1, Employee T2
WHERE T1.EmpTID = T2.EmpTID
GROUP BY T1.EmpTitle
HAVING EmpTitle CONTAINS 'Sales'
```

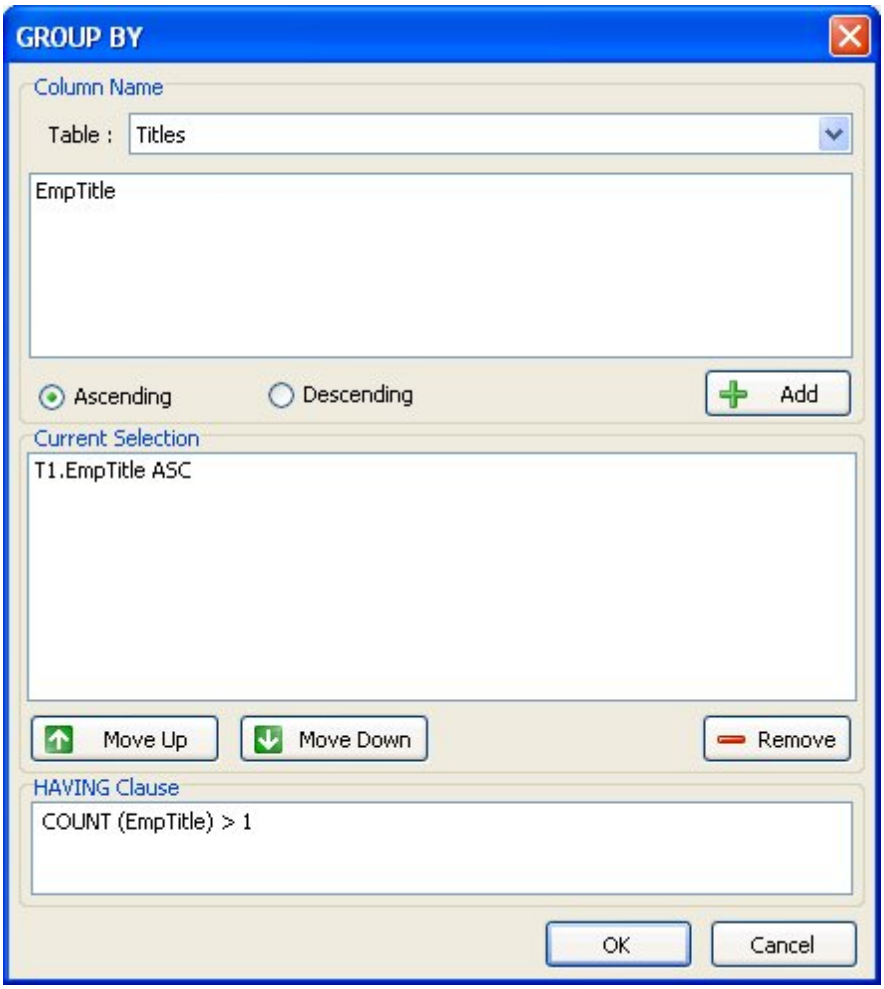
21. Now, browse the query results by selecting "Query" > "Browse Query" from the main Menu Bar. The results should look like the following:



The results should display a column of employee titles that contain the word "sales" in the name, and the number of employees with each title. Close the Data Browser window.

A HAVING clause can also be used to only display limited results based upon employee titles that occur more than once.

22. Right click on the "Titles" table under "Tables/Views In Use", and choose "GROUP BY" from the speed menu.

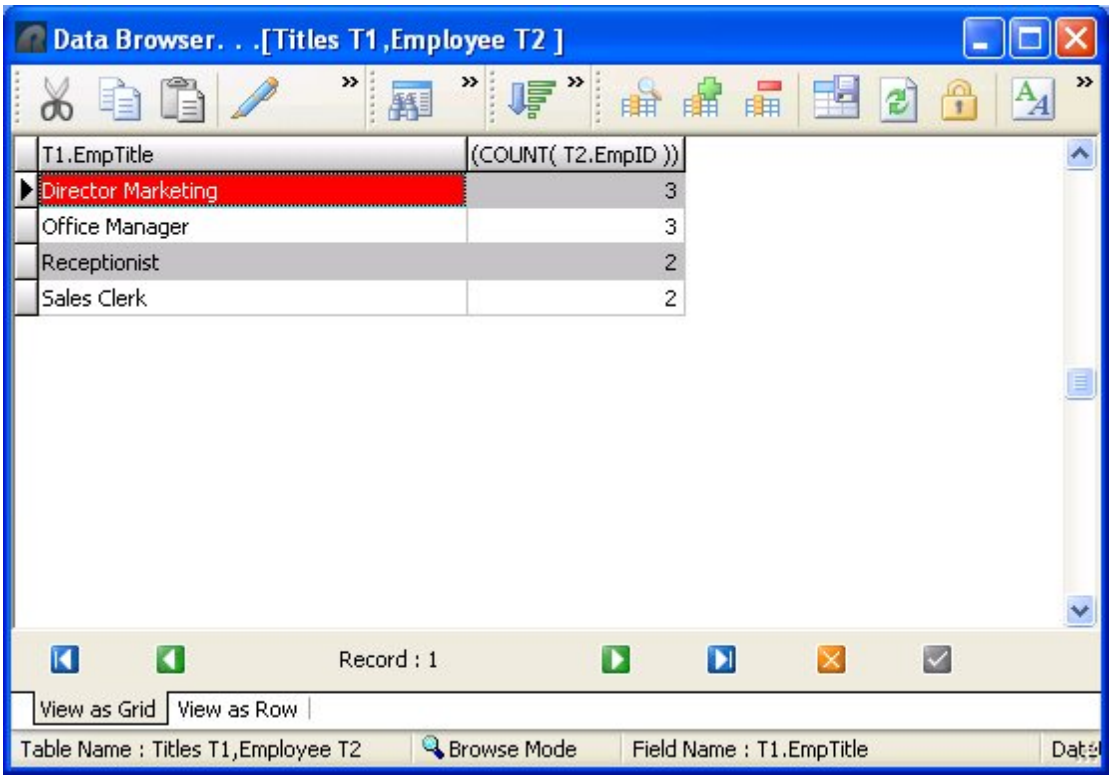


23. Within the "HAVING Clause" panel, enter: COUNT (EmpTitle) > 1
24. Select the "OK" button to return to the Query Builder.

    In the Query Builder main window, you should see the SQL syntax displayed in the bottom of the page as:

```
SELECT T1.EmpTitle,(COUNT( T2.EmpID ))
FROM  Titles T1, Employee T2
WHERE T1.EmpTID = T2.EmpTID
GROUP BY T1.EmpTitle
HAVING COUNT (EmpTitle) > 1
```

25. Now, browse the query results by selecting "Query" > "Browse Query" from the main Menu Bar. The results should look like the following:



The results should display a column of employee titles that occur more than once. Close the Data Browser window.

If you wish to save the query, follow the below steps.

1. From the main Menu Bar, select "File" > "Save Query As View..."
2. In the dialog, enter "JobTitle" within the "View Name:" field.
3. In the dialog, enter "Employee Job Titles" within the "View Comment:" field.
4. Select the "OK" button.
5. Close the Query Builder by selecting "File" > "Close" from the Menu Bar.

In the Database Explorer, the view will be displayed with the name, comment, and number of columns.

Part IV

4 Feedback

Suggestions and Enhancement Requests:

From time to time, everyone comes up with an idea for something they'd like their software to do differently.

If you come across an idea that you think might make a nice enhancement, your input is always welcome.

Please submit your suggestion and/or enhancement request to the R:BASE Developers' Corner Crew (R:DCC) and describe what you think might make a nice enhancement. In R:BASE, the R:DCC Client is fully integrated to communicate with the R:BASE development team. From the main Menu Bar, choose "Help" > "RBG9 R:DCC Client". If you do not have a login profile, select "New User" to create one.

If you have a sample you wish to provide, have the files prepared within a zip archive prior to initiating the request. You will be prompted to upload any attachments during the submission process.

Unless additional information is needed, you will not receive a direct response. You can periodically check the status of your submitted enhancement request.

If you are experiencing any difficulties with the R:DCC Client, please send an e-mail to rbg9rdcc@rbase.com.

Reporting Bugs:

If you experience something you think might be a bug, please report it to the R:BASE eXtreme Developers' Corner Crew. In R:BASE, the R:DCC Client is fully integrated to communicate with the R:BASE development team. From the main Menu Bar, choose "Help" > "RBG9 R:DCC Client". If you do not have a login profile, select "New User" to create one.

You will need to describe:

- What you did, what happened, and what you expected to happen
- The product version and build
- Any error messages displayed
- What computer operating system is in use
- Anything else you think might be relevant

If you have a sample you wish to provide, have the files prepared within a zip archive prior to initiating the bug report. You will be prompted to upload any attachments during the submission process.

Unless additional information is needed, you will not receive a direct response. You can periodically check the status of your submitted bug.

If you are experiencing any difficulties with the R:DCC Client, please send an e-mail to rbg9rdcc@rbase.com.

Part V

5 Useful Resources

. R:BASE Home Page: http://www.rbase.com

. R:BASE eXtreme Home Page: http://www.rbaseextreme.com

. Up-to-Date R:BASE Updates: http://www.rupdates.com

. Sample Applications: http://www.rbasecommunity.com

. General R:BASE Syntax: http://www.rsyntax.com

. Technical Documents - From The Edge: http://www.razzak.com/fte

. More Sample Applications: http://www.razzak.com/sampleapplications

. Education and Training: http://www.rbaseuniversity.com

. Upcoming Events: http://www.rbase.com/events

. R:BASE Beginners Tutorial: http://www.rtutorial.com

Index

Notes