

Building Advanced Queries In R:BASE eXtreme 9.0





Building Advanced Queries In R:BASE eXtreme 9.0

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 9.0

Copyright © 1982-2009 R:BASE Technologies, Inc.

Information in this document, including URL and other Internet web site references, is subject to change without notice. The example companies, individuals, products, organizations and events depicted herein are completely fictitious. Any similarity to a company, individual, product, organization or event is completely unintentional. R:BASE Technologies, Inc. shall not be liable for errors contained herein or for incidental consequential damages in connection with the furnishing, performance, or use of this material. This document contains proprietary information, which is protected by copyright. Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written consent of R:BASE Technologies, Inc. We reserve the right to make changes from time to time in the contents hereof without obligation to notify any person of such revision or changes. We also reserve the right to change the specification without notice and may therefore not coincide with the contents of this document. The manufacturer assumes no responsibilities with regard to the performance or use of third party products.

Products that are referred to in this document may be either trademarks and/or registered trademarks of the respective owners. The publisher and the author make no claim to these trademarks.

The software described in this document is furnished under a license agreement. The software may be used or copied only in accordance with the terms of that agreement. Any unauthorized use or duplication of the software is forbidden.

R:BASE Technologies, Inc. may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from R:BASE Technologies, Inc., the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

Trademarks

R:BASE®, Oterro®, R:BASE C/S:I®, RBAAdmin®, R:Scope®, R:WEB Suite®, R:Mail®, R:Charts®, R:Spell Checker®, R:Docs®, R:BASE Editor®, R:Scheduler®, R:BASE Plugin Power Pack®, R:Style®, R:Code®, R:Struc®, RBZip®, R:Fax®, R:QBDataDirect®, R:QBSynchronizer®, R:QBDBExtractor®, R:Mail Editor®, R:Linux®, R:Archive®, R:Chat®, RDCC Client®, R:Mail Editor®, R:Code®, R:Column Analyzer®, R:DF Form Filler®, R:FTPClient®, R:SFTPClient®, R:PDF Form Filler®, R:PDFWorks®, R:PDFMerge®, R:PDFSearch®, RBInstaller®, RBUpdater®, R:Capture®, R:RemoteControl®, R:Synchronizer®, R:Biometric®, R:CAD Viewer®, R:Twain2PDF®, R:Tango®, R:SureShip®, R:BASE Total Backup®, R:Scribbler®, R:SmartSig®, R:JobTrack®, R:TimeTrack®, R:Syntax®, R:WatchDog®, R:Manufacturing®, R:Merge®, R:Documenter®, R:Magellan®, R:WEB Reports®, R:WEB Gateway®, R:ReadyRoute®, R:Accounting®, R:Contact®, R:DWF Viewer®, R:Java®, R:PHP® and Pocket R:BASE® are trademarks or registered trademarks of R:BASE Technologies, Inc. All Rights Reserved. All other brand, product names, company names and logos are trademarks or registered trademarks of their respective companies.

Windows, Windows 7, Vista, Windows Server 2003-2008, XP, and Windows 2000 are registered trademarks of Microsoft Corporation.

Printed: December 2009 in Murrysville, PA

First Edition

Table of Contents

Part I Building a Query Using a Union	2
1 Quarterly Summary Query	2
1st Quarter	2
2nd Quarter	4
3rd Quarter	5
4th Quarter	6
Browse the Query Results	7
Column Aliases	7
Save the Query as a View	8
Browse the View	8
Part II Building Queries Using Joins	10
1 Inner Join	10
2 Left Outer Join	16
3 Full Outer Join	20

Part



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

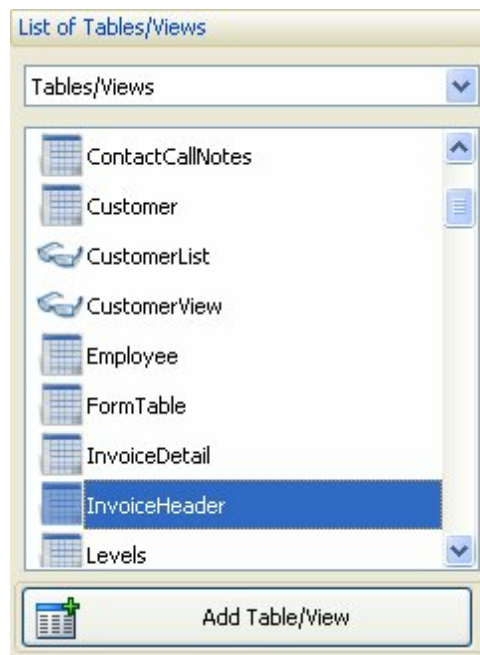
In the RRBWY16 sample database, the "QuarterlySummary" view example utilizes three UNION statements with an initial SELECT statement. The following instructions will step through the process to recreate the view in the Query Builder.

To continue, launch R:BASE eXtreme 9.0 and connect to the RRBWY16 sample database. If you are using R:BASE eXtreme 9.0 (64), the database is located in the following default installation directory: "C:\RBTI\RBG90_64\Samples\RRBYW16". If you are using R:BASE eXtreme 9.0 (32), the database is located in the following default installation directory: "C:\RBTI\RBG90_32\Samples\RRBYW16".

1. Start R:BASE eXtreme 9.0
2. Connect to the RRBWY16 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 "01/01/2009".
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 "03/31/2009".

The Expression panel should have the following displayed:

```
TransDate BETWEEN 01/01/2009 AND 03/31/2009
```

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:

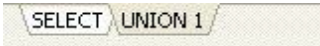
```
SELECT 1,(SUM( T1.InvoiceTotal ))
FROM InvoiceHeader T1
WHERE TransDate BETWEEN 01/01/2009 AND 03/31/2009
```

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 "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 "04/01/2009".
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 "06/30/2009".

The Expression panel should have the following displayed:

```
TransDate BETWEEN 04/01/2009 AND 06/30/2009
```

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 01/01/2009 AND 03/31/2009
UNION SELECT 2,(SUM( T1.InvoiceTotal ))
FROM InvoiceHeader T1
WHERE T1.TransDate BETWEEN 04/01/2009 AND 06/30/2009
```

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 assign the keyword to the WHERE Clause panel.
18. Type in the **third** quarter starting date of "07/01/2009".
19. From the "Operators and Keywords" panel, select the "AND" button to assign the keyword to the WHERE Clause panel.
20. Type in the **third** quarter ending date of "09/30/2009".

The Expression panel should have the following displayed:

```
TransDate BETWEEN 07/01/2009 AND 09/30/2009
```

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 01/01/2009 AND 03/31/2009
UNION SELECT 2,(SUM( T1.InvoiceTotal ))
FROM InvoiceHeader T1
WHERE T1.TransDate BETWEEN 04/01/2009 AND 06/30/2009
UNION SELECT 3,(SUM( T1.InvoiceTotal ))
FROM InvoiceHeader T1
WHERE T1.TransDate BETWEEN 07/01/2009 AND 09/30/2009
```

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 assign the keyword to the WHERE Clause panel.
18. Type in the **fourth** quarter starting date of "10/01/2009".
19. From the "Operators and Keywords" panel, select the "AND" button to assign the keyword to the WHERE Clause panel.

20. Type in the **fourth** quarter ending date of 12/31/2009".
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 01/01/2009 AND 03/31/2009
UNION SELECT 2,(SUM( T1.InvoiceTotal ))
FROM InvoiceHeader T1
WHERE T1.TransDate BETWEEN 04/01/2009 AND 06/30/2009
UNION SELECT 3,(SUM( T1.InvoiceTotal ))
FROM InvoiceHeader T1
WHERE T1.TransDate BETWEEN 07/01/2009 AND 09/30/2009
UNION SELECT 4,(SUM( T1.InvoiceTotal ))
FROM InvoiceHeader T1
WHERE T1.TransDate BETWEEN 10/01/2009 AND 12/31/2009
```

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 01/01/2009 AND 03/31/2009
UNION SELECT 2,(SUM( T1.InvoiceTotal ))
FROM InvoiceHeader T1
WHERE T1.TransDate BETWEEN 04/01/2009 AND 06/30/2009
UNION SELECT 3,(SUM( T1.InvoiceTotal ))
FROM InvoiceHeader T1
WHERE T1.TransDate BETWEEN 07/01/2009 AND 09/30/2009
UNION SELECT 4,(SUM( T1.InvoiceTotal ))
FROM InvoiceHeader T1
WHERE T1.TransDate BETWEEN 10/01/2009 AND 12/31/2009
```

1.1.7 Save the Query as a View

To display the alias values, first save the query.

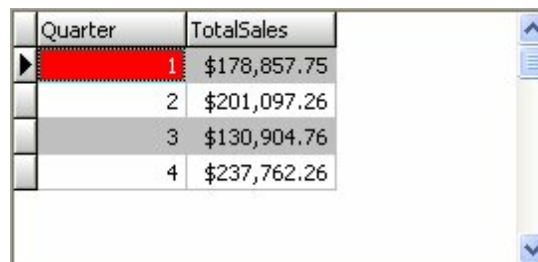
1. From the main Menu Bar, select "File" > "Save Query As View..."
2. In the dialog, enter "QuarterlySales2009" within the "View Name:" field.
3. In the dialog, enter "2009 Quarterly Sales Figures" within the "View Comment:" field.
4. Select the "OK" button.
5. Close the Query Builder by selecting "File" > "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

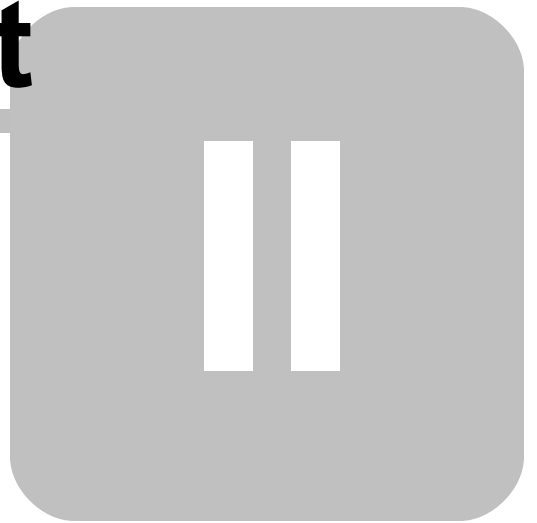
To launch the view, double click on "QuarterlySales2009"

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



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

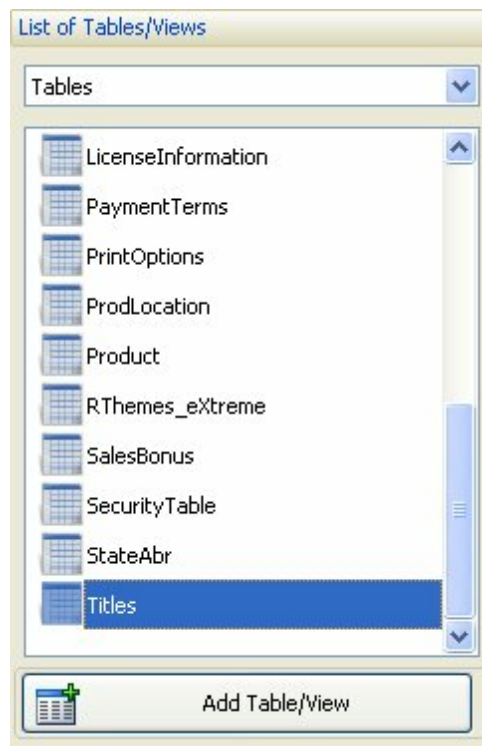
When using an Inner Join, rows are returned when there is at least one row in both tables that match the join condition. Inner Joins are the most common join operation and is the default for R:BASE joins.

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

To continue, launch R:BASE eXtreme 9.0 and connect to the RRBWY16 sample database. If you are using R:BASE eXtreme 9.0 (64), the database is located in the following default installation directory: "C:\RBTI\RBG90_64\Samples\RRBYW16". If you are using R:BASE eXtreme 9.0 (32), the database is located in the following default installation directory: "C:\RBTI\RBG90_32\Samples\RRBYW16".

1. Start R:BASE eXtreme 9.0
2. Connect to the RRBWY16 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" 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.

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 front of the column name.
11. From the "Join Operation" radio button options, choose the equal character (=).

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

Join Properties

Join Type
 No Join Inner Left Outer Right Outer Full Outer

Left Column
Column Name: T1.EmpTID

Right Table/Column
Table Name: Employee
Alias: t2
Column Name: t2.EmpTID

Join Operation
 No operation < <=
 = > >= <>

Preview
INNER JOIN Employee t2 ON T1.EmpTID = t2.EmpTID

OK Cancel

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

Record : 1

View as Grid | View as Row |

Table Name : Titles T1 INNER JOIN Emplc Browse Mode Field Name : T1.EmpTID Datatype :

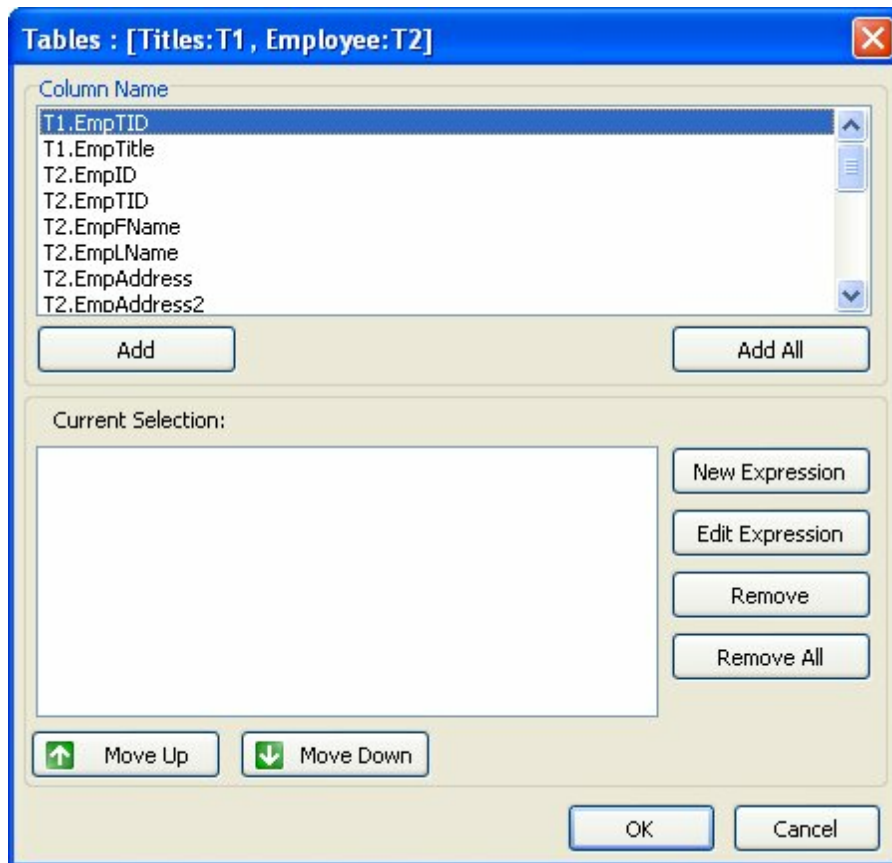
The results show the "Employee" table added to the "Titles" table. Each employee is displayed next to the title where the "EmpID" column matches.

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:

T1.EmpTitle	T2.EmpFName	T2.EmpLName
Office Manager	Jane	Sullivan
Office Manager	John	Minyo II
Office Manager	Sam	Donald
Receptionist	June	Wilson
Receptionist	John	Smith
Sales Clerk	Mary	Simpson
Sales Clerk	John	Chow
Director Marketing	Ernest	Hernandez
Director Marketing	Peter	Coffin
Director Marketing	Darnell	Williams
Director Government Sales	Joe	Donohoe
Outside Sales	Craig	Alan

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

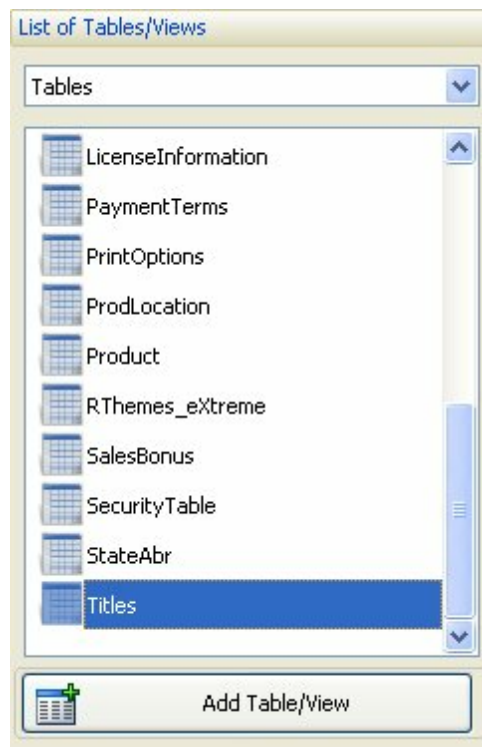
When using an Outer Join, rows are not required to have matching values. The table order in the FROM clause specifies the left and right table. For a Left Outer Join, R:BASE uses each value unique to the left (first) table and completes it with nulls for the columns of the right (second) table when the 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 9.0 and connect to the RRBWY16 sample database. If you are using R:BASE eXtreme 9.0 (64), the database is located in the following default installation directory: "C:\RBTI\RBG90_64\Samples\RRBYW16". If you are using R:BASE eXtreme 9.0 (32), the database is located in the following default installation directory: "C:\RBTI\RBG90_32\Samples\RRBYW16".

1. Start R:BASE eXtreme 9.0
2. Connect to the RRBWY16 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" 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.

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 front of the column name.
11. From the "Join Operation" radio button options, choose the equal character (=).

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

Join Properties

Join Type
 No Join Inner Left Outer Right Outer Full Outer

Left Column
Column Name: T1.EmpTID

Right Table/Column
Table Name: Employee
Alias: T2
Column Name: T2.EmpID

Join Operation
 No operation < <= <>
 [Icon] > >=

Preview
LEFT OUTER JOIN Employee T2 ON T1.EmpTID = T2.EmpID

OK Cancel

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:

T1.EmpTID	T1.EmpTitle	t2.EmpID	t2.EmpTID	t2.EmpFName	t2.EmpLName	t2.EmpAddress
1	Office Manager	108	1	Jane	Sullivan	3935 Old William Penn Highw
1	Office Manager	109	1	John	Minyo II	3935 Old William Penn Highw
1	Office Manager	110	1	Sam	Donald	1715 School Road
2	Receptionist	101	2	June	Wilson	3278 Summit Drive
2	Receptionist	105	2	John	Smith	733 Summit Ave E.
3	Sales Clerk	103	3	Mary	Simpson	12235 Bellevue Ave.
3	Sales Clerk	107	3	John	Chow	5001 Main Street
4	Director Marketing	102	4	Ernest	Hernandez	12390 Windermere Dr.
4	Director Marketing	104	4	Peter	Coffin	4105 29th Ave N.E.
4	Director Marketing	106	4	Darnell	Williams	8806 88th Street
5	Director Corporate Sales					
6	Director Government Sales	111	6	Joe	Donohoe	3935 Old William Penn Highw
7	Manager Support & Services					
8	Outside Sales	115	8	Craig	Alan	3935 Old William Penn Highw

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

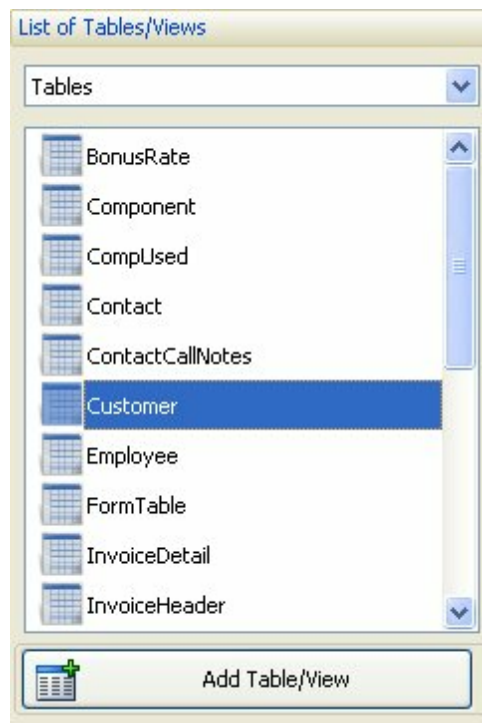
When using an Outer Join, rows are not required to have matching values. The table order in the FROM clause specifies the left and right table. For a Full Outer Join, R:BASE returns all rows, as long as there's matching data in one of the tables.

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

To continue, launch R:BASE eXtreme 9.0 and connect to the RRBWY16 sample database. If you are using R:BASE eXtreme 9.0 (64), the database is located in the following default installation directory: "C:\RBTI\RBG90_64\Samples\RRBYW16". If you are using R:BASE eXtreme 9.0 (32), the database is located in the following default installation directory: "C:\RBTI\RBG90_32\Samples\RRBYW16".

1. Start R:BASE eXtreme 9.0
2. Connect to the RRBWY16 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" 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.

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 front of the column name.
11. From the "Join Operation" radio button options, choose the equal character (=).

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

Join Properties

Join Type
 No Join Inner Left Outer Right Outer Full Outer

Left Column
Column Name: T1.CustID

Right Table/Column
Table Name: Contact
Alias: t2
Column Name: t2.CustID

Join Operation
 No operation < <= <>
 = > >=

Preview
FULL OUTER JOIN Contact t2 ON T1.CustID = t2.CustID

OK Cancel

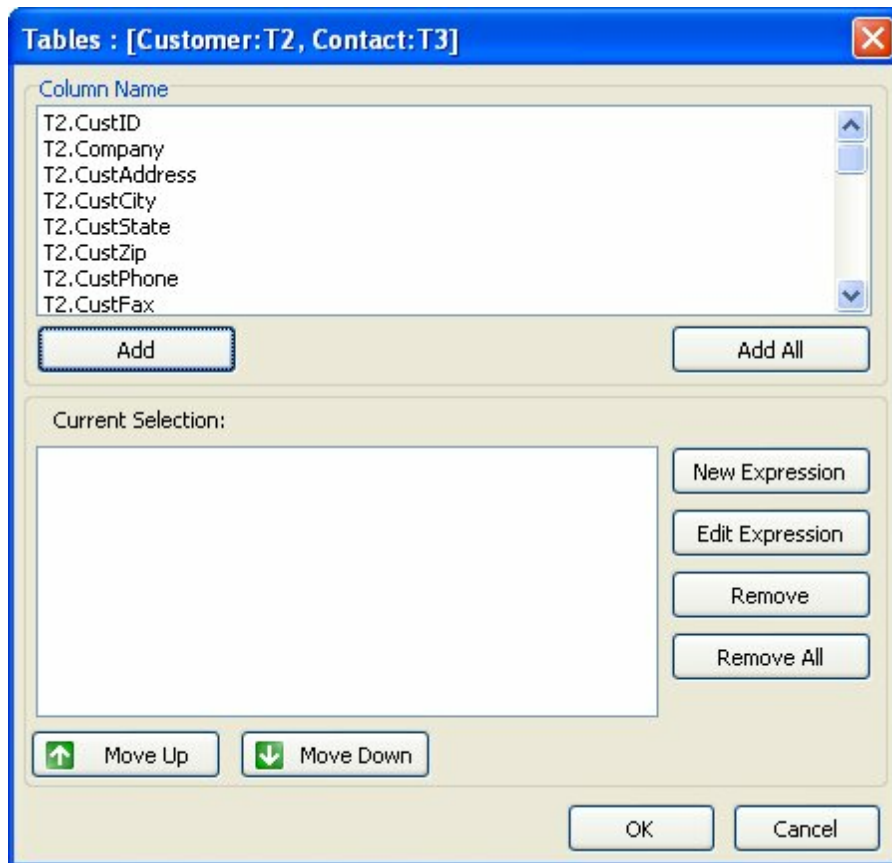
12. Select the "OK" button.

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:

T2.Company	T3.ContFName	T3.ContLName
Computer Warehouse - II	Francis	Sabini
Computer Warehouse - II	Sarah	James
Computer Warehouse - II	Maria	Estwitz
Microtech University - I	Eliot	Preston
Microtech University - I	Walter	Finnegan
Industrial Computers Inc.	Jane	Ferguson
Industrial Computers Inc.	Neal	Bergstrom
Computer Mountain Inc.,	Bill	Stevenson
Industrial Concepts Inc.	Jill	Baileys
Industrial Concepts Inc.	Shelley	Watts
Industrial Concepts Inc.	Arnold	Freeman
PC Consultation And Design	Andy	Chin
Computer Warehouse - I	Bill	Jones
Midtown Computer Co.	Betty	Jones
Midtown Computer Co.	Sharon	Brady
Nordan Distributors, Inc.	Anna	Adams
Nordan Distributors, Inc.	Nancy	Allen
Compdat Computer Consulting	Dennis	Berger
Softech Database Design	George	Clifton
Microtech University - II	Kathy	Dell
Computer Medical Ctr.	Dianne	Peterson
State University	Paul	Frink
Olympic Sales	Jerry	Attwater
Datacrafters Infosystems	Mark	Friedman
Compumasters Computer Supply	Harper	Roy
Data Solutions	Chris	Unger

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.

Notes