

Combining Tables

Peter Vogel



This month, Peter Vogel looks at reporting from two tables where one table overrides entries in the other table.

I have a master table of data that I want to report on. However, I also have a table of daily data. Where there's a matching record in the daily table for a record in the master table, I want to use the daily table's record.

I have a solution that uses a SQL statement, but let me work up to it. I'll start with a query that will show all of your master records that don't have a corresponding record in the daily table:

```
Select tblMaster.*
From tblMaster Left Outer Join tblDaily
On tblMaster.PrimaryKey = tblDaily.PrimaryKey
Where tblDaily.someField Is Null
```

This query joins tblMaster to tblDaily using an outer join—an outer join that forces all of the records from one table into the query, even if there isn't a matching record in the joined table. In this case, because I've used a LEFT outer join, all the records in the table on the left (tblMaster) are forced in. For those tblMaster records where there's no matching tblDaily record, the fields in tblDaily will be set to Null. The Where clause checks for the Null in a tblDaily field and selects only those records where there's no matching tblDaily record.

The query, on the other hand, selects those tblDaily records where there's a matching tblMaster record:

```
Select tblDaily.*
From tblDaily Inner Join tblMaster
On tblMaster.PrimaryKey = tblDaily.PrimaryKey
```

An inner join only finds records where there's a match, so this query pulls out all the daily records where there's a matching history record.

Together these two queries give you the records that you want. To put them together, you use the Union operator:

```
Select tblMaster.*
From tblMaster Left Outer Join tblDaily
On tblMaster.PrimaryKey = tblDaily.PrimaryKey
Where tblDaily.someField Is Null
Union
Select tblDaily.*
```

```
From tblDaily Inner Join tblDaily
On tblDaily.PrimaryKey = tblMaster.PrimaryKey
```

The Union operator requires that the two Select clauses return an identical number of columns, with each column in each position in the Select clauses having the same data type. The SQL processor will use the name of the column in the first query to refer to the columns in both queries.

Most SQL processors will process the first query and then the second query, so the results will be out of order (that is, first all of the tblMaster records with no tblDaily record, then all of the tblDaily records with matching tblMaster records). You can control the order by appending an Order By clause, but the clause must refer only to columns used in the first query (this emphasizes, by the way, that—in the eyes of SQL—what you have here is a single Select query with two Select clauses):

```
Select tblMaster.*
From tblMaster Left Outer Join tblDaily
On tblMaster.PrimaryKey = tblDaily.PrimaryKey
Where tblDaily.someField Is Null
Union
Select tblDaily.*
From tblDaily Left Outer Join tblMaster
On tblDaily.PrimaryKey = tblMaster.PrimaryKey
Order By tblMaster.DateofActivity
```

This will sort both queries together, even though it refers to columns returned by the first query. ▲

 [AA0203.ZIP at www.vb123.com/kb](http://www.vb123.com/kb/AA0203.ZIP)

Peter Vogel (MBA, MCSD) is the editor of *Smart Access* and a principal in PH&V Information Services. PH&V specializes in the design and development for systems that use Microsoft tools. Peter has designed, built, and installed intranet and component-based systems for Bayer AG, Exxon, Christie Digital, and the Canadian Imperial Bank of Commerce. He also wrote *The Visual Basic Object and Component Handbook* (Prentice Hall, currently being revised for .NET). In addition to teaching for Learning Tree International, Peter wrote its Web application development, ASP.NET, and technical writing courses, along with being technical editor of its COM+ course. His articles have appeared in every major magazine devoted to VB-based development, can be found in the Microsoft Developer Network libraries, and are included in Visual Studio .NET. Peter also presents at conferences around the world.