

February, 2004

Advisor Answers

Copying data from one table to another

VFP 8/7/6

Q: What's the easiest way to copy data from a record in one table to a record in another table with exactly the same structure? I've been using SCATTER MEMVAR MEMO and INSERT INTO FROM MEMVAR, but it inserts zeroes in numeric fields that were actually blank in the initial table.

-Frank Cazabon (via CompuServe)

A: Let me start by talking about the issue of blank data. Internally, VFP stores data as character strings, pretty much regardless of the underlying type. (That's not actually the case for some of the newer data types, which use binary representations.) Fields with no specified value are stored as a string of spaces (CHR(32)); such a field is referred to as "blank." When you BROWSE a table, a blank field shows as having nothing in it.

For character fields, of course, that's no different than being empty. For date and datetime fields, blank is the same as empty, as well. However, for numeric and logical fields, an empty field is 0 or .F., respectively, and a blank field looks different than an empty field.

VFP offers a function, ISBLANK(), to check for a blank field. In general, if ISBLANK() returns true, so does EMPTY(), but the reverse is not necessarily true. In addition, the BLANK command lets you restore a field to that pristine condition.

However, when you use the SCATTER command to copy data from a table to variables, blank numeric and logical fields become empty variables. No doubt that's because there's no way to represent the blank value for a variable, since variables use a different internal representation.

If your application depends on distinguishing between blank and empty numeric or logical fields, then SCATTER/GATHER or SCATTER/INSERT isn't a good choice for moving data from one table to another.

COPY TO is one solution if you're creating a new table, but it's no help for adding records to an existing table. SELECT-SQL has the same problem; it creates a new table or cursor and doesn't have the ability to add to an existing table or cursor.

Fortunately, there is a way to copy data directly from one table or cursor to another table or cursor. In VFP 8, it's very simple. The INSERT INTO command can draw its data from the results of a query. You put the query where you'd normally have the VALUES clause or FROM clause.

For your situation, I'll assume that you want to copy the current record in TableA into TableB, and that PK is the primary key field in TableA:

```
m.PK = TableA.PK
INSERT INTO TableB ;
  SELECT * FROM TableA WHERE PK = m.PK
```

The SELECT pulls out the one record we're interested in and uses it as the source data for the INSERT.

Although, it's not what you're looking for right now, it's worth knowing that you're not limited to adding a single record this way or to copying the fields exactly. For example, using the TasTrade data, you might want to track sales of each product on a daily basis, creating a data warehouse of sorts. To do so, you could run this code at the end of each day:

```
INSERT INTO Warehouse ;
  SELECT Product_ID, Order_Date as DateSold, ;
    SUM(Quantity) AS Quantity ;
  FROM Order_Line_items JOIN Orders ;
    ON Orders.Order_ID=Order_Line_Items.Order_ID ;
  WHERE Order_Date = DATE() ;
  GROUP BY 1, 2
```

To test this example, you need to create the Warehouse table first:

```
* In production code, you'd use a table.
* This is a cursor to make clean-up easier.
CREATE CURSOR Warehouse ;
  (Product_ID C(6), DateSold D, Quantity N(15,3))
```

For testing, choose a date that actually occurs in the TasTrade Orders table.

What if you're not using VFP 8? You can still copy data directly, but it takes an extra step. Use a query to put the data you need into a cursor and then use APPEND FROM to add it to the table.

There's one trick needed to make this solution work. APPEND FROM is designed to copy data from tables, not from cursors, so you have to give it the name of the actual table that holds the query result, not the alias for the cursor.

Returning to your situation, you can use code like this:

```
m.PK = TableA.PK
SELECT * FROM TableA WHERE PK = m.PK ;
      INTO CURSOR OneRecord NOFILTER
SELECT TableB
APPEND FROM DBF("OneRecord")
```

This approach works for more complicated cases like the data warehouse, as well:

```
SELECT Product_ID, Order_Date as DateSold, ;
      SUM(Quantity) AS Quantity ;
FROM Order_Line_items JOIN Orders ;
ON Orders.Order_ID=Order_Line_Items.Order_ID ;
WHERE Order_Date = DATE() ;
GROUP BY 1, 2 ;
INTO CURSOR TodaysSales
SELECT Warehouse
APPEND FROM DBF("TodaysSales")
```

-Tamar