August, 1997

ASK ADVISOR

FoxPro 2.x and Visual FoxPro

Q: I am trying to develop a query to perform a distinct count on one field and a distinct sum on a second field (same table). The problem occurs when I join another table (one-to-many). Because the number of rows returned depends on the child table, the first 2 fields of the parent table are replicated in each row. Therefore, I get either an inflated count or sum, whichever is not identified as "Distinct." How can I have both the count and sum distinct? "No Duplicates" has not worked either.

–Mark McCasland (Dallas, TX via Internet)

A: You've diagnosed the problem correctly. Let me explain why you're getting the results you are.

When a query executes, the first thing that happens is that tables are joined in accordance with the join conditions specified, whether you're using VFP5's new JOIN syntax or the older technique of specifying joins in the WHERE clause. Once corresponding records have been matched up, any filter conditions in the WHERE clause are applied. At that point, we have an intermediate result set containing all eligible records.

If there's a GROUP BY clause, it's applied next. All the records in each group are consolidated into a single result record. It's at this point that the aggregate functions (COUNT(), SUM(), AVG(), MIN() and MAX()) are evaluated.

In your situation, the number of records in the intermediate result set for each parent record is the number of children matching the filter criteria. As you note, when you count or sum within a group, the results are skewed. (In fact, the count is the total number of children for the parents in the group.)

Although the DISTINCT keyword can be helpful here, it's tricky. SUM(DISTINCT field) gives you the total of all the different values in the field, so it can eliminate the duplicates caused by the child records. However, if two records in the group have the same value, only one is included in the sum. In addition, as you've noted, you can only use DISTINCT once in a query. So you can't apply it to both a COUNT() and a SUM() (or to two sums or two counts).

The No Duplicates checkbox in the Query Builder (RQBE, in FoxPro 2.x) simply includes the DISTINCT keyword for the query as a whole and is not helpful here.

The problem is really that you're trying to do too much in a single query. Keep in mind that when you're grouping (that is, when a query includes the GROUP BY clause), everything in the field list should be either listed in the GROUP BY clause or involve an aggregate function. If you list any fields where different members of the group have different values, there's no guarantee which member's value is placed in the result. That is, if you write a query like this (using VFP's TasTrade sample data):

```
SELECT City,Country,COUNT(*) ;
    FROM TasTrade!Employee ;
    GROUP BY Country
```

there's no way to know which city will be listed for each country. (In fact, in every version of FoxPro where I've tested this, the chosen value comes from the last member of the group, but you can't count on that always being true.) In fact, because there is no way to predict the meaningfulness of these fields, in some other dialects of SQL (like SQL Server), including non-aggregated fields in a grouped query gives an error.

So, if you need non-aggregate data from the child table, you need to use two queries. The first can work with only the parent table and compute the sum and count and the second query can perform the join and collect the child data needed.

If you also need to use the aggregate functions on the child table, unfortunately, you'll still have to use two queries. Then a third query can consolidate into a single result.

For example, say you want to know, for each TasTrade customer, the number of orders placed, the total freight charges paid, and the total number of items ordered. This series of queries puts the result into a single cursor.

```
SELECT customer_id, COUNT(order_id) AS OrderCount, ;
       SUM(freight) AS FreightTotal  ;
  FROM orders ;
  GROUP BY 1 ;
  INTO CURSOR ParentData

SELECT customer_id, SUM(quantity) AS QuantityOrdered ;
  FROM orders JOIN order_line_items ;
    ON orders.order_id = order_line_items.order_id ;
  GROUP BY 1 ;
  INTO CURSOR ChildData

SELECT ParentData.customer_id, OrderCount, ;
       FreightTotal, QuantityOrdered ;
  FROM ParentData JOIN ChildData ;
    ON ParentData.customer_id=ChildData.customer_id ;
  INTO CURSOR result
```

If you're writing these queries in VFP in order to define a view, all is not lost. You can define three views with the third view based on the other two. When you USE the third view, the first two are automatically USEd. (In fact, the same technique lets you combine data from multiple data sources into a single view. Use remote views to collect data from different back ends, then use a local view to combine the data from the remote views.)

–Tamar