SQL as a Second Language

Thinking in SQL

Learning to think in terms of SQL is a jump for most programmers. Most of your career is spent writing procedural code and suddenly, you have to deal with non-procedural code. The thought pattern has to change from sequences to sets of data elements.

As an example of what I mean, consider a posting made on 1999 December 22 by *J.R. Wiles* to a Microsoft SQL Server website: "I need help with a statement that will return distinct records for the first three fields where all values in field four are all equal to zero."

What do you notice about this program specification? It is very poorly written. But this is very typical of what people put out on the Internet when they ask for SQL help.

There are no fields in a SQL database; there are columns. The minute that someone calls a column a field, you know that he is not thinking in the right terms.

A field is defined within the application program. A column is defined in the database, independently of the application program. This is why a call to some library routine in a procedural language like "READ a, b, c, d FROM My_File;" is not the same as "READ d, c, b, a FROM My_File;" while

"SELECT a, b, c, d FROM My_Table;" and "SELECT d, c, b, a FROM My_Table;" are the same thing in a different order.

The next problem is that he does not give any DDL (Data Definition Language) for the table he wants us to use for the problem. This means we have to guess what the column datatypes are, what the constraints are and everything else about the table. However, he did give some sample data in the posting which lets us guess that the table looks like this:

```
CREATE TABLE Foobar
(col1 INTEGER NOT NULL,
 col2 INTEGER NOT NULL,
 col3 INTEGER NOT NULL,
 col4 INTEGER NOT NULL);

INSERT INTO Foobar
VALUES (1, 1, 1, 0),
       (1, 1, 1, 0),
       (1, 1, 1, 0),
       (1, 1, 2, 1),
       (1, 1, 2, 0),
       (1, 1, 2, 0),
       (1, 1, 3, 0),
       (1, 1, 3, 0),
       (1, 1, 3, 0);
```

Then he tells us that the query should return these two rows:

```
(1, 1, 1, 0)
(1, 1, 3, 0)
```

Did you notice that this table had no name and no key specified? While it is a bad practice not to have a declared PRIMARY KEY on a table, just ignore it for the moment.

At this point, people started sending in possible answers. *Tony Rogerson* at Torver Computer Consultants Ltd came up with this answer:

```
SELECT *
  FROM (SELECT col1, col2, col3, SUM(col4)
          FROM Foobar
```