While I've come to really like our new reporting tool FastReports, there are a few things that took some creativity to figure out. One of those things were "Many to Many" - or m:n - relations. Here's how it's done.

Background

A typical example would be a list of orders. On one side you've got an "Order" table with general information like customer, date, etc. On the other side you've got your "Products" table with item name, cost per unit, etc. And in between there is your "OrderPositions" table that puts the products into the orders.

Orders **OrderPositions** **Products**

- orderid
- customerid
- orderdate
- ...

- posid
- orderid
- productid
- amount

- productid
- description
- unitcost
- ...

The intuitive ways, that don't work

As someone who already made a few reports with "1 to Many" - or 1:n - relations my first intuition was to simply modify that approach. Where in 1:n you've got your master Data Band and a nested Data Band for the details I tried to reappropriate the nested Data Band for the products while keeping the master Data Band for the orders. The relations between the tables where properly defined so that should work, right? Nope

I figured that leaving a step out might confuse the tool. So my second approach was to nest a third Data Band inside the other nested Data Band. The top Data Band would hold the orders, the middle one the order positions and the bottom one the products. Now surely with every step explicitly defined that should work... Wrong again.

How it works

Ironically I started out with the best possible groundwork and headed off in the wrong direction. Only two Data Bands are necessary. The top one belongs to the orders and the nested one to the order positions. In the fields displaying the data one now has to follow the relation from the middle table to the desired outer table.

Instead of trying to place:

[Products.description]

on the nested Data Band the correct expression would be:

[OrderPositions.Products.description]

Why it works like this

A nested Data Band can only be at the "Many" side of a relation to its parent Data Band. It's purpose is the display multiple tuples related to a specific parent tuple. And this wasn't give between the middle and bottom Data Band in my first attempt. For every order position there is never more than one product - another nested Data Band doesn't make sense.

Conversely for every order position there is never less than one product either. That's why you can directly place the related fields of the "Products" table in the Data Band of the "Order Positions" table.