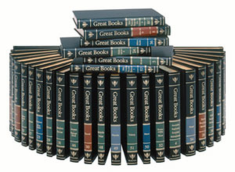
In the last article, we talked about a very common issue of synthetic keys in the QilkView data model and discussed its various resolution techniques. In this article, we gonna talk about another common scenario which leads to a bad data model and causes certain performance issues while setting up the data model, which is **Circular References** in QlikView.

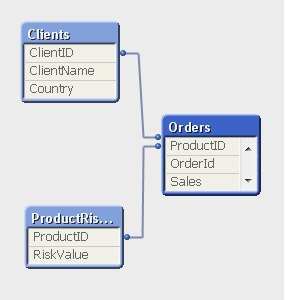
[](https://munishbansal.files.wordpress.com/2014/11/circular-reference-1.png)

 The good thing about Qlikview data model is that tables get automatically connected to each other based upon certain key names in there, called as associative data modelling. This feature is very useful, but produces certain structures which are not considered good & degrade the report performance. One of them is these unwanted structure is known as circular reference. This logical loop i.e. circular reference occurs when there is two or more path of association among three or more tables i.e. from one table to another can be reached via multiple paths.

Let’s consider an example where we have three tables

a.      Clients – ClientID, ClientName, ClientLocation AS Country  
b.     Orders – OrderID, ClientID, ProductID, Sales  
c.      ProductRiskValue – ProductID, RiskValue

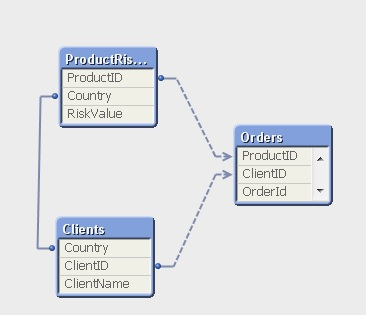
Data model will look something like:

[](https://munishbansal.files.wordpress.com/2014/11/circular-reference-1-1.jpg)

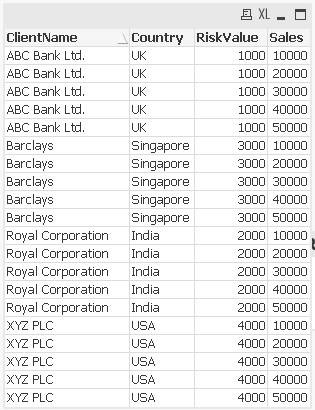
This looks fine as of now. Let’s add ‘**Country**’ column in the ProductRiskValue table.  
When we load these set of tables, below will be prompted:

[](https://munishbansal.files.wordpress.com/2014/11/circular-reference-2.jpg)

 This warning clearly specifies two things, 1. Due to the circular reference / loop this may result the ambiguous results and 2. Qlikview will automatically handle the situation by setting one or more tables as loosely coupled table(s). Look at the data model:

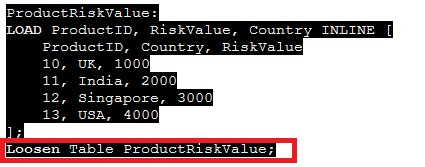
[](https://munishbansal.files.wordpress.com/2014/11/circular-reference-3.jpg)

 Here, relation between Orders & Clients tables is via ClientId whereas you can get from orders to the ProductRiskValue either directly via ProductID or via the Clients tables (ClientID, Country) and therefore a circle is formed among these three table which is called as circular reference.

As seen in the table view, Orders table is appearing as loosely coupled table which means there can be ambiguous results around it. Let’s put the fields into a table box:  
[](https://munishbansal.files.wordpress.com/2014/11/circular-reference-4.jpg)

Sales for (example) ‘ABC Bank Ldt.’ is reported wrong because ideally this client is associated with UK which in turn to ProductID=10 & hence sales of only 10000 should have been listed there.

More about Loosely coupled table – As seen, QlikView marks one of the tables as “loosely coupled”, which means that the logical inference cannot propagate through this table. Qlikview usually sets the loosely coupled table as the largest table in the loop (most often a fact table). However this automatic behavior can be overridden by a setting in the script as below:

[](https://munishbansal.files.wordpress.com/2014/11/circular-reference-5.jpg)

How to deal with the circular reference in Qlikview, solutions:

•        When you get a circular reference ask yourself if you could live without one instance of the field that is causing the extra association (such as a duplicated field). **If you can, rename it or remove it.**  
•        Otherwise you may have **to resort to concatenation or a link table to remove**the circular reference.

1.      Rename the fields causing the circular reference – if we know (e.g.), country in the product table is actually the ProductCountry rename the field so that client-country & product-country to not infer the same thing.

2.     Concatenating the two tables: If we are sure that two fact tables are more or less have the same kind of data, these can be concatenated together into one table and hence avoid the circular loop. But in our case Orders & ProductRiskValue are of completely different significance. In this case Link table helps.

3.      Using Link tables – Link tables can be used for linking two or more tables (fact tables) where we have fields (dimKeys) with the same names – measures may be of different meaning across Fact tables.