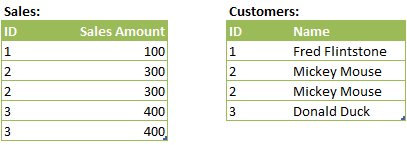
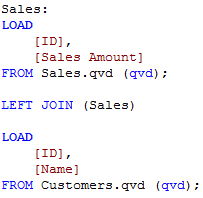
The case

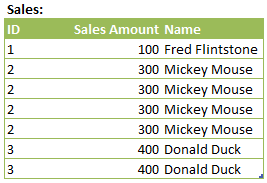
Consider the following two tables; **Sales** and **Customers**:



You will notice that both tables contain duplicate records, the **Sales**table contains identical records for ID’s *2* and *3*, and the **Customers** table contains two records for *Mickey Mouse*. If we JOIN these two tables together using the following script:



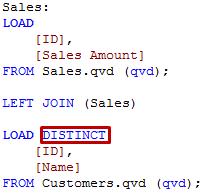
Then the result is the following table:



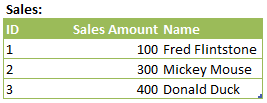
There are no surprises here. There are 4 rows for *Mickey Mouse* with a Sales Amount of *300*, as both tables contained two duplicate rows and 2 x 2, the number of ways in which both records can be joined, equals 4. There are also 2 rows for *Donald Duck* with a Sales Amount of *400*, as the **Sales** table contained duplicate rows for ID *3*.

The DISTINCT LOAD

Now we add one small change to the script, we use LOAD DISTINCT to load the second table:

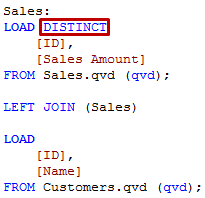


Since I already revealed the spoiler at the start of this post, that using LOAD DISTINCT with the JOIN or CONCATENATE statements will also remove all duplicates from the target table, you have probably guessed that the previous piece of script results in the following table:

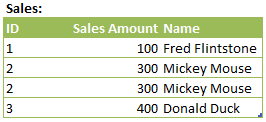


The twist

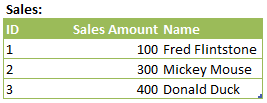
Now that we’ve seen how LOAD DISTINCT impacts the resulting table, let’s make one small change to the script; instead of using LOAD DISTINCT on the second table we’ll use it on the first table:



Can you guess what the resulting table is going to look like? Think about it for a second before scrolling down further. When deduplicating the first table and then joining the second table (which also contains duplicates), the result must be something like the one shown below, right?



In an M. Night Shyamalan-esque twist (ok, so not really), the result is actually exactly the same as when we used LOAD DISTINCT on the second table:



### The conclusion

Whenever the DISTINCT clause is used on a table, the table’s records will stay distinct, no matter what you JOIN or CONCATENATE to the table. This can have all sorts of unwelcome consequences, such as duplicate-but-correct facts disappearing from your fact table. Of course, it can also have positive effects; this behavior is very useful when you need your link table to only contain distinct combinations

### The advice

Be aware of this behavior in QlikView, if you need to do a DISTINCT LOAD combined with a JOIN or CONCATENATE statement consider if you might be losing duplicate rows, and if this is something that is desirable. If it’s not, then you should probably load the data into a temporary, resident table before joining it to your other data.