Troubleshooting Query Performance Issues

When a query is executed, it can sometimes seem to take an unusually long time before you see any results, or the running of the query may “time out” giving you an error message. (Do note that after initially building a query, the first time it is run will be slower than on subsequent attempts.)

Most of these situations can be prevented through careful structuring of the query and attention to detail. This includes testing output incrementally, the choice and ordering of tables/records, a logical use of joins, the selection and ordering of criteria, the economical utilization of fields, and the use of expressions with Wes functions. At the end of this article are links to documentation with further detail on query performance.

If you have a query that is performing slowly, timing out or returning an unmanageable number of records, here are some items to look at.

1. **Table Size/Complexity** - If the query eventually completes running, albeit slowly, take a look at the tables used and consider if it makes sense for the process to be slowed because of the size of the underlying tables and the complexity of the request.

2. **Restart and test the Query, incrementally** – To test the query create it anew, one step at a time, i.e., begin with one table, if feasible, and test some criteria to see if the information returned is reasonable, and once you are satisfied, add the next table.

3. **Number of Tables** - Note that the fewer the tables that you use as part of your query, the better.

4. **Table Order and Joins** - Examine the order in which you bring the tables into the query. It is important to realize that the order in which tables are added to the query affects the joins that are produced.

5. **Join on Related Records** - The quality of the joins can have a pronounced impact on the results. If at all possible, join on related records. If there are no fields that match up, you will return either no data or a very large amount of meaningless data (a Cartesian join).
6. **Criteria Order** - Another important aspect of the ordering of elements is to be sure that the criteria are sorted properly. If, for example, in your original query you added a criterion related to the Record A *after* criteria related to Record B, the query needs to process all the rows in Record A before it can reach Record B.

7. **Amount of Criteria** – Remember that the more you put in for Criteria, the quicker the query will run.

8. **Descriptions from Other Tables** - Check to see if you are asking for translate table values or description fields from other tables; if so, this will add more joins and more overhead. Where possible, use a *Wes Function* to supply a description.

9. **Number of Fields** – Reduce the number of fields in your query as much as possible.

For more information, check these pages:

- [Query Tip Sheet](#)
- [PeopleSoft Query – Advanced Concepts, Lesson 11](#)
- [Wes Functions](#)