**How To and Not To Use Sub-Queries:**

* If you are going to use a sub-query, put your filters in the sub-queries if you can.
* Correlated sub-queries in the SELECT list are a REALLY bad idea – generally speaking. Engage your brain and figure out how to rewrite that as a sub-query that you JOIN in the FROM clause.
  + Sometimes, the query optimizer can essentially turn it into a join, but I find it best not to count on that.
  + However, a correlated sub-query MIGHT make a better alternative to using a CURSOR in applicable situations.
* You may need to create a new version of the VIEW you are using that applies additional filter criteria… or just copy the guts of that VIEW into your SQL code to give you more control
  + The downside to this is that if the logic of the VIEW changes, you have to change your query or other version of the VIEW, too.
* Don’t overcomplicate your sub-queries/views. This is often easy to do when you are using a view or sub-query that someone else wrote – they might include extra joins and fields that you don’t need.
* A VIEW is a sub-query… it is NOT a stored execution plan and will be incorporated and recalculated within the context of your query.
  + However, using VIEWs can cause table redundancy in your main query.
  + If the data from joined tables in the VIEW are not needed to either 1) get the VIEW to return proper results (i.e. as a filter) or 2) get values from other tables needed in your consuming/main query, then your VIEW will be costing you extra overhead.