Using python to speak with SQL Server data.

There are two options:

1. Use the native ArcSDESQLExecute module. This requires no installation by the customer as it ships with ArcGIS. It requires a .sde connection file to be created (this can be created by hand with an ArcGIS Toolbox tool when the package is delivered – doing this is code is probably not worthwhile as once it is created it will be there forever. It is a simple procedure).

The module will do the following:

* Connect to an SQL database (done, works OK)
* Run queries and extract non-spatial data (done, works OK)
* Run a stored procedure (done works OK).

It does however not extract the spatial data held in the SP\_GEOMETRY column in the SQL server database (all queries attempting to access this column fail without error message). I have spent some time seeing if there is a workaround, but there isn’t:

*"If you are trying to export geographic data as geographic data from an enterprise geodatabase, ArcSDESQLExecute is the wrong tool. The tool doesn't work with native spatial data types returned from various DBMSes."*

This was from <https://geonet.esri.com/thread/169418>

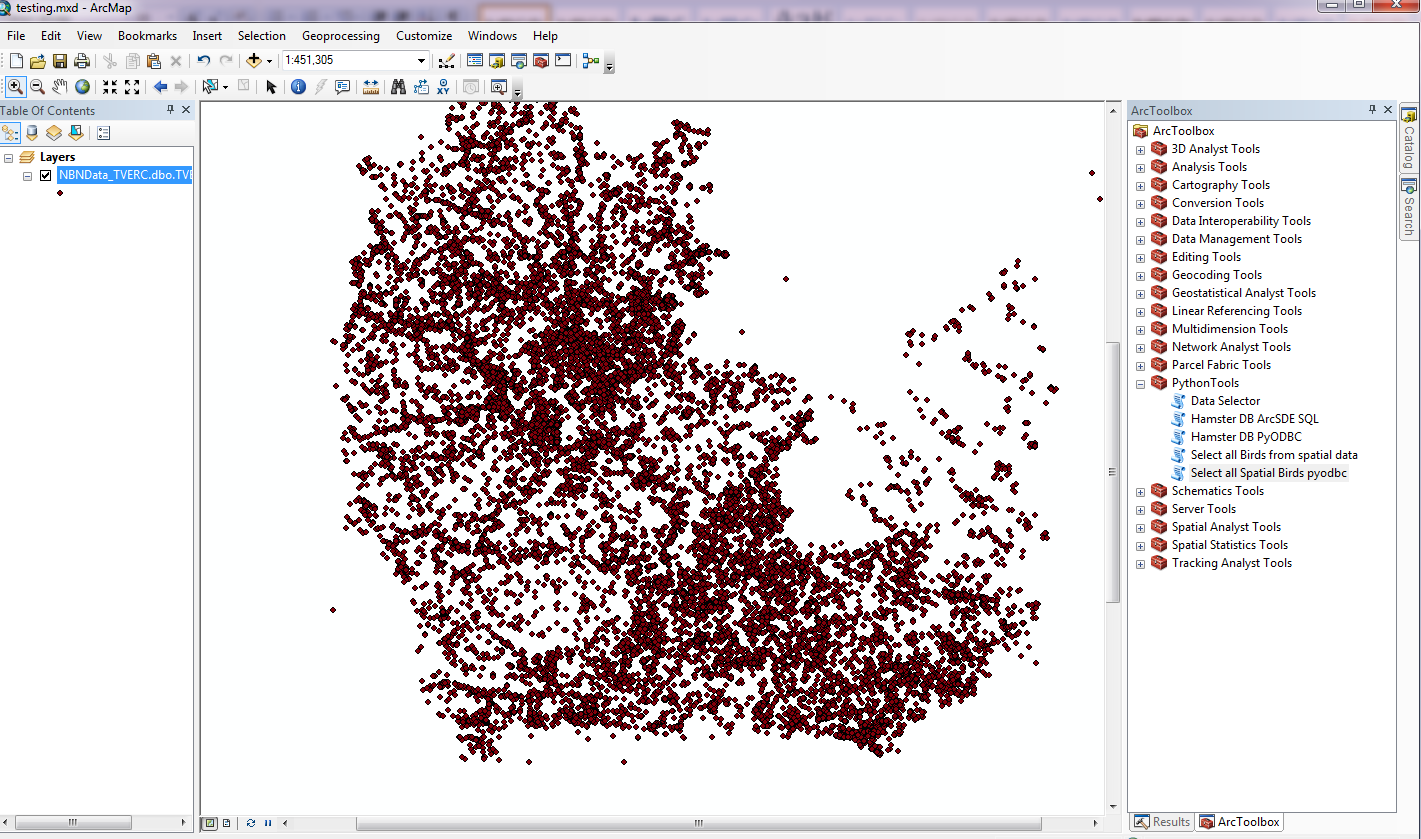
It basically means that it will work, as long as we use the X and Y fields contained in the data, not the geometry (which I believe is points only anyway, so it does not add a great deal of value).

1. Use PyODBC.

As discussed before, this does require the installation of pyodbc on the user’s computer. It is marginally easier to use than the ArcGIS solution. It does all the above, but also extracts the information in the SP\_GEOMETRY field. However I have not been able to convert the Microsoft Geometry object to Shape – there must be a conversion but this is certainly not easily available (other than going through a third-party piece of software which defeats the object).

In conclusion – I think we are good to work with either of the two approaches and I’m happy to pursue the native solution. Given that our geometry objects contain only points, it should be OK to re-create the shapes using the X and Y co-ordinates.

As an aside, it is very odd that ArcGIs will attach to the SQL Server db and use the geometry field to plot up the data (see picture below). If it becomes ever necessary to use the native geometry (e.g. they start storing polygons in it, or similar) it can be done using a Make Query Table / Query Layer call – however this would involve calling the SQL from within the script, not as a stored procedure. It’s another option that can be explored, I’m not sure how quick it might be.



Geometry data straight from SQL Server