1. The TourneyMachine process should start by going to the homepage and parsing all data within the javascript “tournaments” variable.
2. It should then take this raw data and insert it into the staging table within the clubstats database

Azure Servername is “clublacrosse.database.windows.net”

uname = sql\_admin,

Pwd = DCdc4646

Schema = “stg” (as in “staging”)

SELECT TOP 100 \* FROM [clubstats].[stg].[tourneymachine\_mainpage\_data]

We stage this file in the above table. We then have a separate process to parse this file for loading into another table. Dates like ‘Jan 1 – Jan 10, 2019’ are converted to 2 date variables ‘start\_date’ and ‘end\_date’.

The final table is [data].[tm\_mainpage\_data]. Data passed into the second python script (to spawn spiders to get the game results should only be when the games are in the past (< GETDATE()) and we have not already loaded the scores. We will have a separate process that will set the ‘is\_active’ = 0 once game scores are loaded. In this case, the game\_id from the link is the key to seeing if data exist).

1. The second python script should take the ‘link’ parameter from an ODBC call to the processed data table, download those endpoints, parse the scores, and create a CSV. We will then insert that data into another staging table that will prepare to process the results.

SELECT link

FROM [data].[tm\_mainpage\_data] WHERE

1=1

AND end\_date < GETDATE()

AND is\_active = 1

ORDER BY end\_date