**Updates to Database**

Changed name of what was pInsertWorkHistory to pInsertFullWorkHistory...Then deleted pInsertWorkHistory as it was redundant.

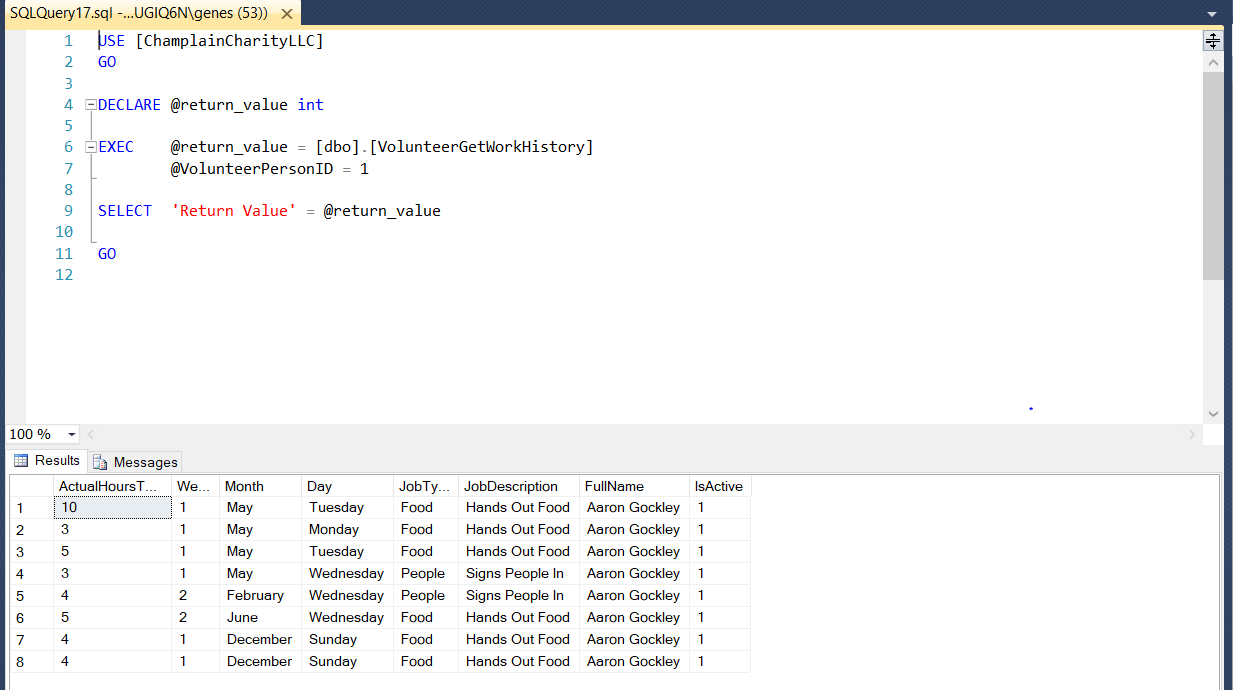
Altered plan to instead of being one massive stored procedure be a series of smaller stored procedures. I will go into more detail in a later section.

Added the pInsertVolunteerJobs into pInsertFullWorkHistory and all other insert work history procedures (details in later part)

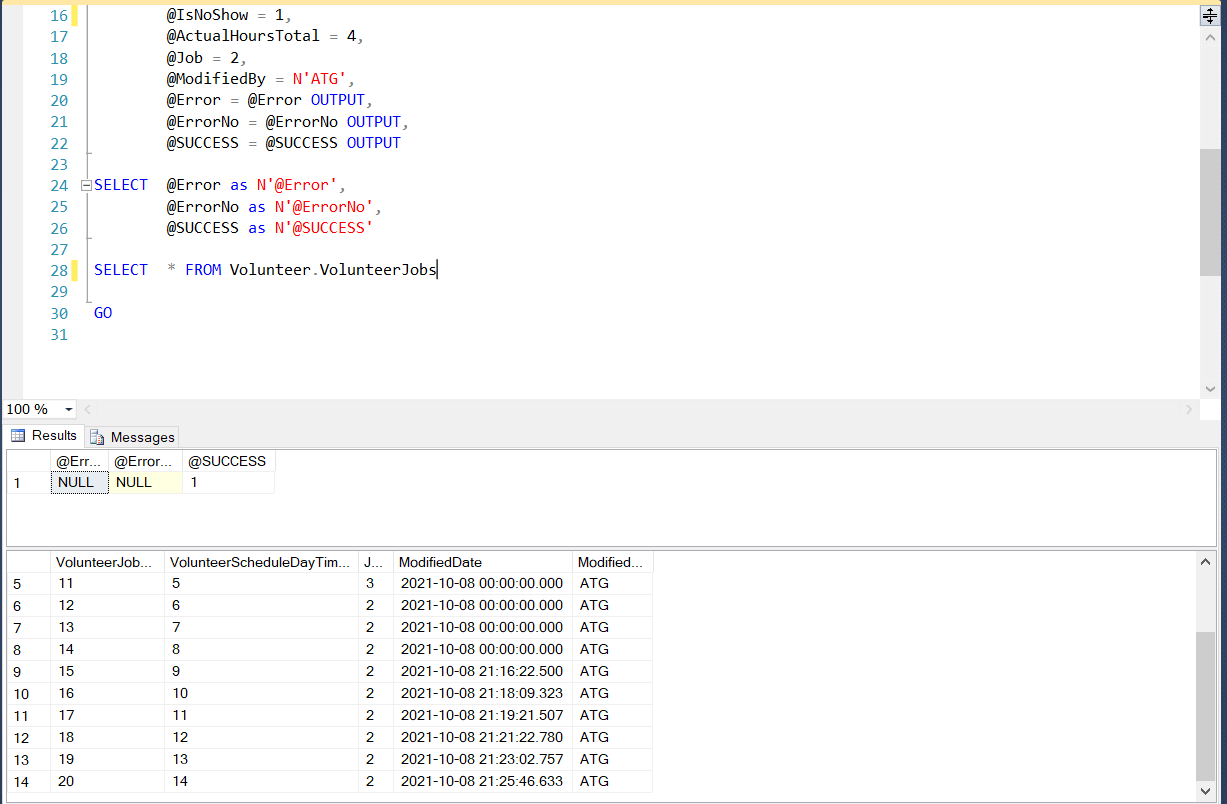
Added Jobs to GetWorkHistory Function, it calls up the JobType and JobDescription from LUVolunteerJobType for readability.

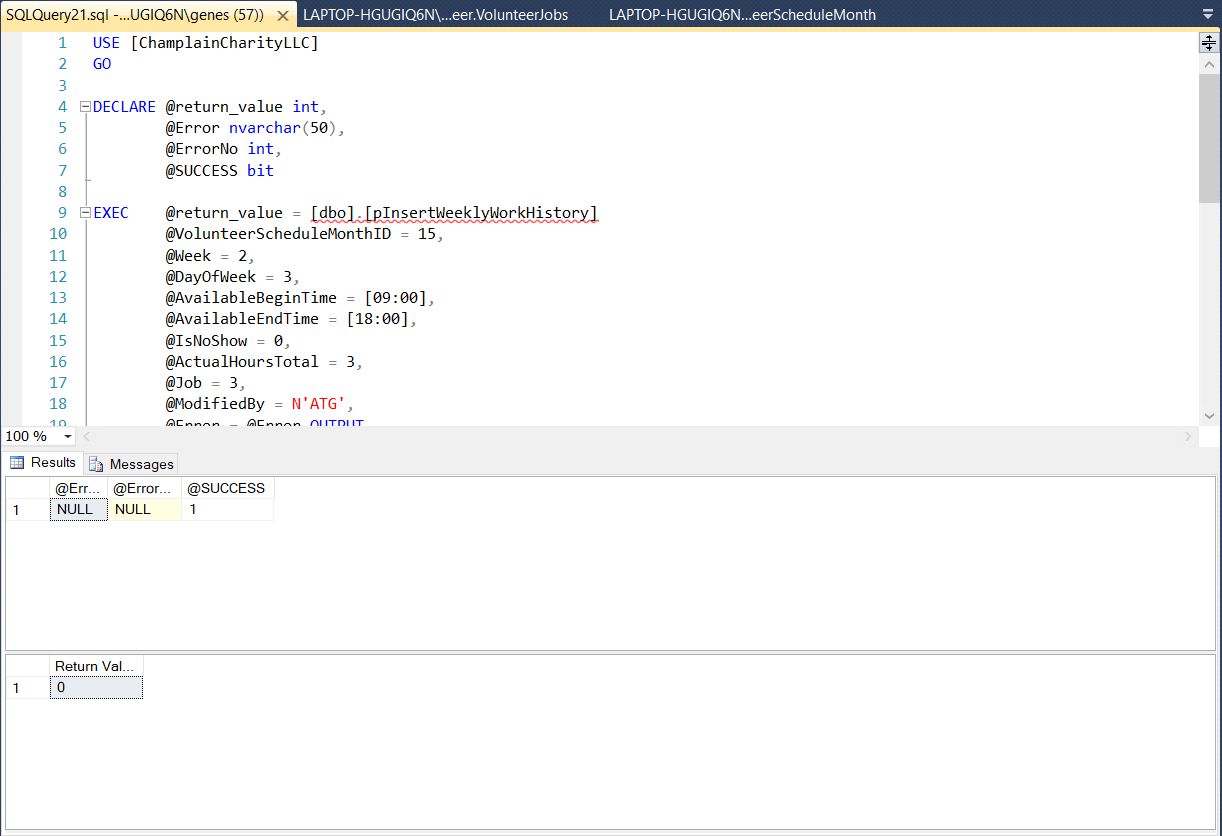
Added those two columns to the Procedure that uses the Function.

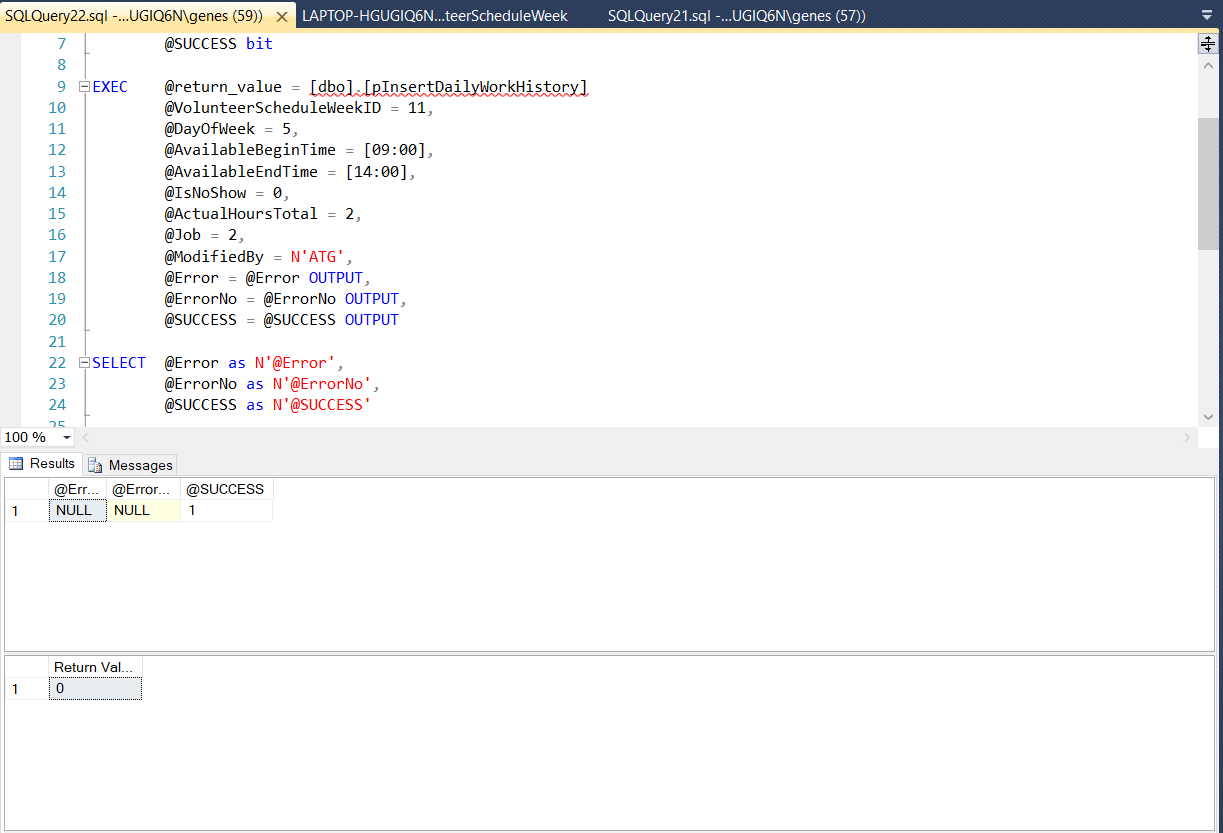
Added data to Volunteer.VolunteerJobs to reflect the changes to the driver insert procedures. Retroactively added jobs data using each volunteerdaytimeID.

**Screenshots**

The above screenshot shows the VolunteerGetWorkHistory Procedure running, you can see that the output below shows that it works.

Here is where my plan changed. My initial attempt was to make a massive stored procedure with a series of if statements dictating which set of stored procedures would be used. I tried for a bit, but it became clear very quickly that it was not working. It was also becoming a massive unusable mess. I might have been able to make a driver stored procedure that calls each of the three driver procedures I just made, but I wanted to keep this sample. This one inserts data in the Month, Week, Day, and Jobs table. 

This one is for entering more than one week in a particular month for someone. You enter the Month ID and then the rest of the information. 

This procedure allows you to insert multiple days in a week that a person works. This fulfills the requirement that a person can work multiple times in a week, but does not need to work every week.

**Project Discussion**

I am happy with where I am at with this project. The only change I want to make is a way to extract information from Person.Person and insert it into Volunteer.VolunteerContactInformation. I think it would not take very long. I will try and make it for next week. I have the scheduling done. The biggest challenge I faced with the scheduling was realizing that my initial plan was far too complicated for what was necessary for this project. I dialed back my plan and I am very happy with how it ended up. The way I want it to work now is I want it to call up the different procedures on the front end. All of these are nice and easy to understand as well as easy to read. Readability and simplicity is something I always try to do with my code.

I have my UDF finalized, as well as the procedure that uses it. I have my scheduling logic procedures made, and all of them work. They took some time to get working, but they now work. I realized I accidentally joined the wrong columns and something completely broke, but I fixed it. All I have left is a document explaining how this works, collecting all my testing documentation together, as well as combining all of my changes into one document. There might be some changes depending on how the week goes.