# Charlie Tarnoff

# 2/21/21

# IT FDN 130

# Assignment06

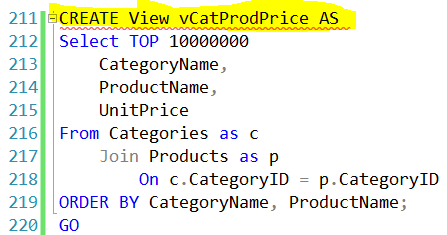
# Module 6: Views

# Introduction

This paper will focus on the similarities and differences between Views, Functions, and Stored Procedures.

# When to Use a SQL View

A view is a user-created table that is based off one or more tables in a database. Views are typically saved SELECT statements but can also include WHERE, ORDER BY, and JOIN statements. Views are useful because they act as an abstraction layer over a table. This allows changes to be made to underlying data tables while not effecting the end-user who is querying data using a view. Another reason to use a view is it is easier to set permissions for certain tables that might contain sensitive data. Figure 1 shows an example of a view being created:

\

***Figure 1: An example of a view being created.***

# Differences and Similarities Between a View, Function, and Stored Procedure.

Functions or “User-Defined Functions” act similarly to views but have a couple key differences. Functions can return an individual value (i.e. integer, date) whereas views cannot. While the syntax between the two is fairly similar, there must be a “Returns Table” included in the code after the “Create Function” code to signify that a function is being used instead of a view. Another difference is there must be “dbo” placed in front of the table you are creating a function on.

A Stored Procedure is a named set of SQL statements. These statements can be simple or as complex as needed but it is more common to use stored procedures for complex reporting. Stored procedures are very similar in syntax to a view but the main difference between the two is when you query a stored procedure you do not use a SELECT statement, but rather an EXECUTE statement.

# Summary

I found Module 6 to contain very practical and useful information in regards to writing SQL code that has very realistic real-world applications. The ability of views to create abstraction layers from tables in a database has immense benefits from both an end-user and back-end development perspectives.