Sidney Hutchison

February 19, 2024

IT FDN 130 A: Foundations of Databases & SQL Programming

Assignment 06

<https://github.com/SidHutch/DBFoundations>

**SQL Views**

# **Introduction**

In this document I will discuss Views in SQL, including when to use a View compared to when you may instead want to use a Function or a Stored Procedure.

# **When to Use a SQL View**

A View is a saved Select statement, which can be simple or complex.

Best practice is that every table in your database should have a Base View, which is essentially just a View that shows the table exactly as the table is on its own. You would do this so that you can force people to use the View instead of accessing the original table. This is helpful in situations where you want to make changes to the original table, as you can just update the View so that applications that other developers have already made using that View will not be affected by the table being different than it was previously. So, you should use a View whenever you make a new table.

Views are also often used when you want the Public to see only a portion of a table. For example, if a table includes columns with private, sensitive information, you do not want just anyone to be able to see those columns of information, so you would create a View that does not include those sensitive columns. Then, when you deny public access to the original table and instead grant public access only to the view, people would only be able to access the view and they would not be able to see that sensitive information.

Another benefit of using Views is the Schema Binding option. This will prevent anyone from accidentally changing the original table too much if it would stop the View from working. This is helpful for trying to mitigate silly mistakes that could have big consequences.

# **Similarities Between a View, Function, and Stored Procedure**

Views, Functions, and Stored Procedures are all named sets of SQL statements, and are used for saving your SQL statements in the database instead of just in your script file. You can actually create a separate View, Function, and Stored Procedure that all do the exact same thing, though the syntax of the code when creating each of those would be a bit different.

# **Differences Between a View, Function, and Stored Procedure**

One of the differences between a View, Function, and Stored Procedure is potentially what each one of these tools will return to you after executing it. A View returns a table. A function can return a table if you specify it to do that using “Returns Table”, but it can also return an individual value. Stored Procedures are very flexible. You can have many statements within a Stored Procedure, and they do not need to just be Select statements.

Another difference between these three tools is the syntax when you are writing them out, as well as their use of parameters. You need to write “dbo.” ahead of your Function’s name for it to work, which is not required for a View nor a Stored Procedure. A Function also has parentheses at the end of its name, which is not something you include for a View, and is only sometimes included for a Stored Procedure if you want your Stored Procedure to have parameters. Functions can use parameters in their parentheses to change the results of the query as it is executed. This is often done with a Where clause that involves the parameter being in the Function somewhere. You can also specify a separate Where clause when using a View though, so this is not as big of a difference as one might initially think.

# **Summary**

SQL Views, Functions, and Stored Procedures are all tools that can be used in both similar and different situations to save sets of SQL Statements in your database, instead of just in your script. Sometimes you can use them interchangeably depending on what statement(s) you are wanting to save, but other times you will want to use a particular one of these tools as they do have several differences which might be important in your specific circumstance.