Views, Functions, And Stored Procedures

## Introduction

When SQL Select statement become more and more complex it sometimes becomes more efficient to save those Select statements within a database’s file as a SQL View, Function, or Stored Procedure. As opposed to saving your statements on a hard drive, or a SQL Script, Views, Functions, and Stored Procedures allow you to store your code on the database. When utilized properly, these can enhance security, allow you to split data, restrict access to data/tables, make complex queries easier to reuse, and more.

## When To Use SQL Views

There are several examples of when it is best practice to us a SQL View. A view can be used to extract data for reporting purposes, and that is called a Reporting View. When Select statements in your reporting become more and more complex, you can save your complex Select statement by creating a View and that View can now be used in a much simpler Select statement making the code easier to use for future use. Another time to use Views would be for specialized reporting. For instance, if you only want a table to show certain rows and/or columns. You can split your data by rows (horizontal partitioning) with Views by using a Where clause. You can also divide your data in columns with a View and grant or deny access to certain tables or parts of tables.

Another good time to use a View is whenever you create a Table. Each table in a database should be able to show data from a table while still restricting access to the table and this is done by creating a Base or Basic View. This allows people to still use your data, but in the “Abstract”, and prevents people making changes to your original tables by restricting access to them. It’s best practice for each table in a database to have a Base or Basic View.

## Views, Functions, And Stored Procedures

Similar to Views are Functions and Stored procedures. Custom Functions (sometimes referred to as User Defined Functions or UDFs) come in two basic types of Functions that can return a table of values and that return a single value. They are like Views in they can save complex Select statements, but unlike Views, Functions can use parameters. Although Views can simply use a Where clause for similar results when returning a table of values, Functions can be very useful when returning a single, or scalar, value as an expression. Views are not able to return a scalar value as an expression. A good application for this would be in conjunction with a Check constraint. This would allow you otherwise would not be able to reference a column in another table.

Stored Procedures are also similar to Views and Functions in that they are a Named Set of SQL Statements and are another way to store data in a database file. However, unlike a View or Function where you would Select the code, Stored procedures Execute the code. This means it runs the code, as opposed to Views and Functions use Select and treat it as a table. Stored procedures are very flexible as well as you can do many statements within a Stored Procedure, and not just Select statements. You can have Inserts, Updates, Deletes, etc., and create tables in a Stored Procedure. This can be a better route for more complex reporting.

## Conclusion

When SQL Select statement become more and more complex it sometimes becomes more efficient to save those Select statements within a database’s file as a SQL View, Function, or Stored Procedure. While there are differences between Views, Functions, and Stored Procedures, they are all a Named Set of SQL Statements and are another way to store data in a database file. Views, Functions, and Stored Procedures allow you to store your code directly on the database, and when utilized properly, they can enhance security, allow you to split data, restrict access to data/tables, make complex queries easier to reuse, and much more.