Ethan Morss

February 24, 2021

IT FDN 130 A Wi 21: Foundations of Databases & SQL

Assignment 06

Views Functions and Stored Procedures

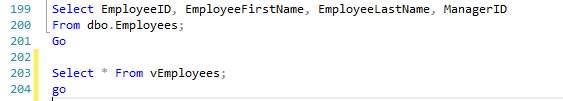
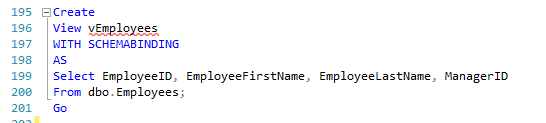
# Introduction

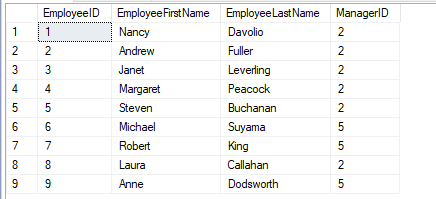
In SQL we use to Views Save select statements and report information out of a database. This paper will examine the utility obtained from a Views, who might benefit from them. Additionally, it will examine some Functions and Stored Procedures, and how they are similar to and different from Views.

# Utility

In SQL, a view is a Select statement that is stored in the database. As simple as that sounds, a View offers a great deal of utility and makes databases easier to work with for both coders actively working with the database, and for passive users who need limited access to information from the database but should not be, or are uninterested in, working in the deeper details of the database itself. To understand the utility derived from the View we need to understand the differences and similarities between a standard select statement, and a View.

Figure 1.1 shows a select statement that is not stored, and a base View from the Employees table which is stored in the database. Figure 1.2 shows the View when it was created; it shows the full statement being engaged when the View is selected; the syntax and logic are nearly identical to the Select statement. Figure 1.3 shows that the data each statement returns is identical; they return the same results.

Figure 1.1Figure 1.2

 Figure 1.3

To the active user of the database, the view saves time. At its creation, the view statement may take more energy to set up as it is a bit of an invention but recreating that statement by typing all the characters and logic behind it every time it is needed uses time and increases the likelihood for error.

To the passive user of the database, the view provides access and simplicity. Views make it possible to expand access to other users of the database; this could be either be someone who needs regular reports but does not have a lot of experience making statements or understand the logic of databases, or a Sr. Manager who knows a number of languages but needs to check in on project quickly.

# Minimizing Contact with Data and Providing the Latest Data

Other benefits from using the view, are that the view does not interact with data, it recalls it, and displays it; the information returned is a representation of the data, not the data itself. A user cannot use a view statement to update or alter the database. The view is an abstraction, and as such denies passive users a chance to alter things accidentally.

As a side, it should be observed that database managers can limit access to actual tables and provide access to views which makes the source data of databases more robust.

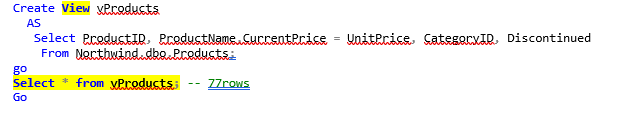
Additionally, the view captures the latest data entered, so if someone enters data to the database moments before a user enters the view, it is returned by the view.

# Differences between Views, Functions, and Store Procedures - Theoretical

There are similarities and differences between Views, Functions, and Stored Procedures. At a basic level, each can be made to return the same thing, which his important to remember as we often do not want to over complicate things with our queries. However, when constructing more advanced statements the differences between the three become more important.

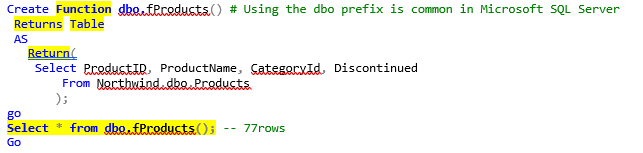
Since the first part of this document focused on Views, and our lectures have only covered minimal aspects of Functions and Stored Procedures, rather than compare and contrast the three in detail, I’ll provide a brief list of the differences and similarities, along with an image showing how they vary in syntax and execution.

## The View

Figure 2.1

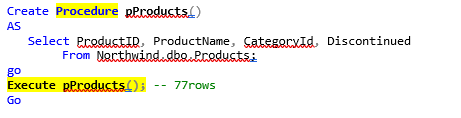
The view does not accept parameters, can contain only one select query, can not call stored procedures but can call functions, returns a single table, can not handle exceptions, can be used in a select query, and should return a table.

## The Function

Figure 2.2

The Function accepts parameters, can contain several statements, cannot call stored procedures, can return a single value or table, can not handle exceptions, allows one select statement, can be used in a select query, and should return a value.

## The Stored Procedure

Figure 2.3

The Stored Procedure accepts parameters, can contain several statements, can call functions and views, can return multiple values and tables, exceptions can be managed with a try catch block, cannot be used in a select query, and its not mandatory that it returns a value.

# The Differences between Views, Functions, and Store Procedures - Application

Given my lack of experience making more complicated SQL statements, I struggle describing the differences between the three. The list above suggests how these three operations might be used differently, but its beyond my experience to describe in applications.

At this point, I’m comfortable saying Views are the simplest and should be relied on as often as possible, as simplicity offers many benefits.

Functions are able to use an argument and pass through a value to return another value. In that sense, Functions rely on a dynamic variable that is given some treatment by the greater statement; this power seems compelling enough to make one endure its complexity.

Lastly, Stored Procedures, seem to be the most powerful and complicated of three in that they can be used to update the database and insert new, or modify, existing data.

Until I get more hands on exposure to the three in applications, I won’t be comfortable describing the differences between the three futher.

# Summary

In summary Views are saved select statements attached to a database. A view creates takes less effort to execute than a Select statement, and is less error prone, it offers benefits to people who need a lot of familiarity with a specific database. Also stored in a database, are Functions and Stored Procedures, which can produce similar results as View. While Views are simpler than Functions and Stored Procedures, the options they offer for modifying tables, data, arguments, and parameters make them worth exploring further.