Erin Leggett
IT FDN 130-A
Module 06 - Assignment 06
August 18, 2021

Basic Abstraction: Views, Functions, and Stored Procedures

Objective

Write a one-page document responding to the following prompts about SQL views.

Explain when a SQL view would be used.

A view is the result of a stored dynamic SQL query which, once defined, can be recalled for a number of purposes and ends. As views are not included in the physical schema of the database, they allow users to interact with data in such a way that the source data is not altered - this as a concept is referred to as *abstraction*. Abstraction plays a vital role in maintaining data integrity. Furthermore, the results of a SQL view are not recorded as such; only the view itself occupies space on the server. This allows both administrators and users alike to store and recall complex queries - enabling a further layer of abstraction between the user and the syntax of the SQL query as well. Let's say our marketing team regularly requests electronic mailing lists to communicate with our active user base. Instead of granting the marketing team full access to our customer database, we could easily create a view to select and display only active users and their email addresses. Not only would this prevent the marketing team from making changes to the information in our user database, which could have dire consequences; it would also restrict them from viewing and accessing both raw data and SQL code that is irrelevant to the task at hand.

Explain the similarities and differences between a View, Function, and Stored Procedure.

Views, functions, and stored procedures all serve the same rudimentary aim: to simplify end user interaction with the database by storing and defining commands that reliably produce a predetermined result. Not only does this eliminate the need for data to be stored in several instances across a database; it also allows for certain elements to be recalled and employed repetitively, therefore simplifying further production efforts. In many cases the same set of results can be produced by a view, function, and stored procedure; however, one's ability to interact with and manipulate that end result depends on which of the three they are using.

The differences between these abstraction tools, then, lie primarily in their overall complexity. The results of a view are static and will return the same table regardless of how and when they are queried, save the ability to employ a basic WHERE clause. Functions and stored procedures, however, are more dynamic: as they both refer to a group of stored SQL statements, they are far more customizable and can interact with parameters. Even then, functions are limited compared to stored procedures: while functions by nature cannot alter source data, stored procedures can execute INSERT/UPDATE/DELETE statements in addition to SELECT. Furthermore, both a view and a function by definition must return a value of some kind - although, unlike views, the result of a function can be scalar (i.e., a single value). That is not necessarily the case with a stored procedure, which can return a null or even no result, depending on precisely what is being executed.

Resources

Microsoft, 2017, *Stored procedures,* accessed 17 August 2021,<https://docs.microsoft.com/en-us/sql/relational-databases/stored-procedures/stored-procedures-database-engine?view=sql-server-ver15>.

Microsoft, 2017, *Views,* accessed 16 August 2021,<https://docs.microsoft.com/en-us/sql/relational-databases/views/views?view=sql-server-ver15>.