Sam Colgan

Feb. 22, 2023

IT FDN 130

Assignment 06

https://github.com/samcolgan/DBFoundations.

Views in SQL

Introduction

One of the main objectives of SQL programming is to write queries that are easy for users to navigate and access. With this objective in mind, it is helpful to create views that consolidate complex Select statements into singular objects. Similarly, SQL queries can benefit from functions and stored procedures, which are similar to views in that they hide multiple lines of code behind an abstraction layer, but differ in that they can be used to report more than just Select statements.

Using Views

Anytime a SQL programmer writes a Select statement, they should consider capturing that Select statement as a view. According to Tutorials Point, "A view is nothing more than a SQL statement that is stored in the database with an associated name." (www.tutorialspoint.com/sql/sql-using-views.htm.) (External Site). In other words, views store the code that is used to call the table, but they do not store the table itself. Views can be separated into base views—which represent each newly created, unmodified table—and reporting views, which represent tables that have been modified from the base tables in some way. The advantage of using base views is that they can preserve the integrity of the data in the original table by granting access to the view but not to the table. Furthermore, base views are useful for constructing Application Programming Interfaces (APIs) since web applications that pull data from the table will break if the data in the table is modified, but will not necessarily break if the application pulls from the view. Meanwhile, reporting views are helpful for making complicated Select statements more accessible to novice users, who can access the view instead of replicating the Select statement. Anytime a programmer wishes to use a table more than once, they should consider using a view.

Other Kinds of Abstraction Layers

In addition to views, there are other kinds of abstraction layers like functions and stored procedures. All three are used to express batches of SQL code as singular objects. Of the three, views are the most narrowly defined since they only capture Select statements. Functions are more broadly defined since they can input a parameter value and output multiple object types, such as a scalar or a table. Stored procedures are the most broadly defined of the three since they are simply batches of SQL code. Unlike views and functions, stored procedures are executed rather than selected. Finally, views, functions, and stored procedures are all stored in SysComments.

Summary

As established, the main utility of abstraction layers like views, functions, and stored procedures is that they consolidate complicated code batches into singular objects. Whereas views are only applicable for returning tables, functions work for returning both tables and single values, and stored procedures work for returning any batch of SQL code. Hence, views are the most narrowly defined, stored procedures are the most broadly defined, and functions are in between the two.