Nicholas Mei
August 06, 2024
IT FDN 130 A Su 24: Foundations Of Databases & SQL Programming
Module 06: Views – Assignment 06
Github URL: https://github.com/nmei-42/DBFoundations

SQL Views, Functions, and Stored Procedures

Introduction

In this assignment, I will be explaining what a SQL view is and when someone would want to use one. Then I will be contrasting the similarities and differences between a SQL view, function, and stored procedure.

SQL Views

First, before explaining its use cases, it is necessary to explain what a SQL view is. A SQL view is a virtual table whose rows and columns contain data defined by a SELECT statement (a.k.a. a query)[1,2]. Views have a multitude of uses, that can be quite handy. A view can be used to present a streamlined or simplified view of one or more tables for a database user[1]. A view can also be used as a means of access control by limiting reads to the views and preventing user access to underlying core tables[1]. Finally, views can be used as a backwards compatibility layer to present an emulated version of table prior to a schema change that adds/changes columns[1].

SQL Functions

SQL functions (often also referred to as a User-defined functions or UDFs) work like functions in other programming languages[3]. They take in parameters, perform calculations/actions, and return the result[3]. In Microsoft SQL Server, there are three broad categories of functions, scalar functions, table-valued functions, and system functions[3]. Scalar functions return a single value of a specific data type (e.g. int, float, date, money, etc.)[3]. Table-valued functions in contrast, return a “table” data type that is the result of a SELECT statement defined within the function. Finally, system functions are built-in functions that can’t be modified, but provide various features ranging from functions to help parse JSON to basic mathematical operations like a function that returns the square root of an input value[3,4]. Functions can help modularize and deduplicate code, and can have faster execution (via caching) and network traffic reducing properties[3].

SQL Stored Procedures

SQL stored procedures are a grouping of one or more SQL statements that can take in input parameters and can return multiple outputs in the form of output parameters, can call other procedures, and return a status value indicating success or failure[5]. Stored procedures have many benefits. They can reduce server/client network traffic because procedures are stored server-side, they can improve performance because they don’t require a new query plan each time they are called, and they can improve security because permissions can be setup to allow running a stored procedure but not to read or interact with underlying database objects (like tables)[5].

Contrasting Views, Functions, and Stored Procedures

Views, UDFs, and stored procedures can be used to accomplish similar results, especially if all that is needed is to SELECT and display data from one or more tables, however they have very critical differences[2,3,5,6]. Views are limited compared to UDFs and stored procedures because they only support SELECT statements, can't take input parameters, and don't "return" anything[1,2]. UDFs are less limited than Views in that they can return scalars or tables, but they can't perform SQL statements like UPDATE, INSERT, and DELETE on non-local (to the function) tables (whereas stored procedures can)[3,5]. A user can call a function within a stored procedure but cannot call a stored procedure within a function[6]. Stored procedures in general support a larger number of SQL statements and clauses compared to UDFs[3,5]. These include things like TRY…CATCH blocks as well as transactions[3,5].

Summary

In summary, I've described how SQL views can be used to streamline presentation of a table, limit access, or maintain compatibility after a schema change/migration. I've also described UDFs and stored procedures and compared their similarities and differences.

[1] markingmyname, rwestMSFT, WilliamDAssafMSFT, Jak-MS, KevinConanMSFT, alexbuckgit, nachoalonsoportillo, rothja, Illuminae, julieMSFT, MikeRayMSFT, cawrites, Kat-Campise, CarlRabeler, WilliamAntonRohm, v-maudel, akost, v-alje, craigg-msft, … BYHAM. (2023, May 23). *CREATE VIEW (Transact-SQL) - SQL Server*. Microsoft Learn. https://learn.microsoft.com/en-us/sql/t-sql/statements/create-view-transact-sql?view=sql-server-ver15

[2] *SQL - Using Views*. (n.d.). Tutorialspoint. Retrieved August 5, 2024, from https://www.tutorialspoint.com/sql/sql-using-views.htm

[3] rwestMSFT, alexbuckgit, rothja, niko-neugebauer, JKirsch1, MikeRayMSFT, avborup, julieMSFT, pmasl, craigg-msft, & Saisang. (2024, July 29). *User-defined functions - SQL Server*. Microsoft Learn. https://learn.microsoft.com/en-us/sql/relational-databases/user-defined-functions/user-defined-functions?view=sql-server-ver15

[4] markingmyname, huypub, rothja, arvindshmicrosoft, rwestMSFT, julieMSFT, cawrites, DCtheGeek, MikeRayMSFT, craigg-msft, MashaMSFT, Saisang, edmacauley, & barbkess. (2023, May 23). *What are the Microsoft SQL database functions? - SQL Server*. Microsoft Learn. https://learn.microsoft.com/en-us/sql/t-sql/functions/functions?view=sql-server-ver15

[5] WilliamDAssafMSFT, rwestMSFT, rothja, DCtheGeek, Court72, john-par, damabe, MikeRayMSFT, craigg-msft, & Saisang. (2024, January 5). *Stored procedures (Database Engine) - SQL Server*. Microsoft Learn. https://learn.microsoft.com/en-us/sql/relational-databases/stored-procedures/stored-procedures-database-engine?view=sql-server-ver15

[6] Calbimonte, D. (2017, February 14). *Functions vs stored procedures in SQL Server*. SQLShack. https://www.sqlshack.com/functions-vs-stored-procedures-sql-server/