Raul Guajardo

02/23/2021

Foundations of Databases and SQL Programming

Assignment06

[Raul’sGitHub](https://github.com/RaulGuajardoOrtega/DBFoundations)

# Views

Introduction

In this week’s paper we will be discussing SQL Views, Functions and Stored Procedures. A View is a type of virtual table that basically consists of a **Select** Statement with SQL code stored, which at the same time is stored inside the Database.

A Stored Procedure is a group of SQL statements compiled into a single execution plan. It can also be used for Select Statements, but is more known to perform actual changes to tables.

Functions are subroutines that contain one or more SQL statements. They are similar to Views as in they can return data from tables, but they are limited when it comes to make actual changes to the Database. Essentially, Functions have Read-Only access.

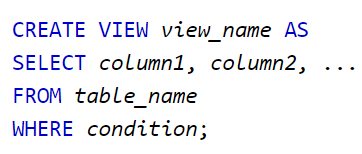
When to use a View?

There are many reasons to why use Views instead of working directly with the data contained in our Database. The one that I consider the most important has to do with protection of user data and privacy.

As we know, when working with Databases there will be multiple users connected to it at the same time. The data stored in the Database would normally be valuable, both for the business and for the customer. By creating Views, we allow users to visualize the data in “virtual” tables, therefore, we have the responsibility and capacity of deciding who is able to see what. Financial details, personal information such as a SSN are two prime examples of data that we must protect from users that either don’t have the credentials to access to that data or don’t have the responsibility of doing so.

A second reason for using Views is that sometimes the users that have access to the Databases have no deep understanding of SQL, for example a Marketing team, therefore, it would be helpful to write queries behind the virtual table, kind of like working in the backstage, to only present the information that is relevant for them.

A third reason is to prevent users to make accidental changes to the Databases. Building Databases take time and effort, and we want to ensure that they remain consistent so that users and applications remain functional. Figure 1 shows the code to create a View.



*Figure 1 shows the code for creating a View.*

Differences and similarities between Views, Functions and Stored Procedures.

As mentioned before, a View allows the user to visualize a virtual table. Most of the times a View will contain a **Select** statement specifying the data the query will return. In addition to **Select**, we can also use Joins and Subqueries as well as other SQL statements such as **Where** and **Join**.

If we are looking to make actual changes to a Database, a View will not be an option, since it only allows Select statements. A Stored Procedure will be an option, as it allows changes by using statements such as **Insert**, **Update** and **Delete**. These changes can be made to a single or multiple tables.

When it comes to performance, a View will perform fundamentally identical to a Stored Procedure, as this [study](https://www.scarydba.com/2016/11/01/stored-procedures-not-faster-views/)(www.scarydba.com/2016/11/01/stored-procedures-not-faster-views/) shows.

Earlier we discussed some of the benefits of using Views. The key benefits of using Stored Procedures are precompiled execution, reduced client/server traffic, efficient reuse of code, programming abstraction and enhanced security controls.

A Function is very similar to a Stored Procedure, the major difference is that a Function must always return a value, but a Stored Procedure may or may not return a value.

Unlike Views, where the results are presented and can’t be changed (a Where clause can be used to solve this), a Function can use parameters to change the results of the query. Figure 2 shows a Function.

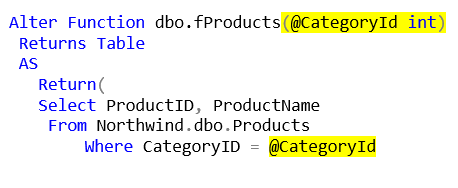


Figure 2 shows a Function being executed to return data.

Summary

There are different approaches and functionalities when working and manipulating data from Databases. Depending on the level of abstraction, reliability or functionality required, we can choose to work with Views, Stored Procedures or Functions, or a combination of these. The advantages that these bring to the table are evident, and we must understand their differences so that we can make better choices.