Robert Boyd

8/15/2021

ITFDN130

Assignment 06

https://github.com/CaldwellAzvian/DBFoundations

**Views, Functions and Stored Procedures in SQL**

**Introduction**

This week, we learned how to implement views, functions and stored procedures in SQL, which can benefit us significantly when designing a database for use by people other than ourselves as a database developer. In the following paper, I will discuss when we would use a SQL view, and what the similarities and differences are between views, functions and stored procedures.

**When Do We View?**

Views in SQL are effectively stored select statements that can be selected from like you would a table. One reason we would use a view is that we can use it to shorten a query we are doing multiple times, especially if that query is extremely complex, which allows us to contain it into a single statement like you would a table for ease of use. However, its primary use and benefit comes from adding an abstraction layer between the user and the database’s tables. An abstraction layer allows design changes of the database to be made significantly easier and maintain the way that the data is queried by existing infrastructure, like an application, if the underlying database is changed. A great example would be like that of an employee name field. If employee name was one column in the database originally, and then that was queried directly from the table, if we were to ever change the base table to first and last name being separate columns, anything that queried the table directly would cease to work and require modification directly to every use case. If we had a view that targeted the table instead, we could make that underlying change to the database, but we could then also alter the view to combine first and last name together with a space in between. This would result in existing applications being able to continue operating without having to make changes in every single place the select was called because they were all utilizing the view which as far as the user is concerned remained unchanged. Ultimately, we would want to use a view when we have a very complex select statement that we will be using multiple times to simplify code or if we want to add an abstraction layer.

**Views, Functions and Stored Procedures**

As previously explained, a view is effectively a stored select statement, where a long and complex series of instructions can be shortened to a single line, and it adds an abstraction layer for a user. However, two other named sets of SQL statements that we have are functions and stored procedures.

A function is very similar to a view in that it can have a series of instructions contained within it, and they are both something that can be selected from. However, a function is different in primarily two ways: first, they can be provided parameters for their operation, which may not seem as valuable when they are used as a reporting function to select a table because a view can be used similarly with the use of a where clause. The second way they are different however makes this first difference important: a function can return either a table or a single value in something called a scalar function. This means that they can take in parameters and perform calculations with those parameters and then return a single value, allowing them to perform complex calculations with a simple call and fed in parameters which makes them distinct from views.

Stored procedures are again, similar to the other two in that they represent a named set of SQL statements that can be called with a single line. However, they are very different in that they are not selected from, they are executed and this difference may seem small, but has large implications: they do not necessarily have to return or select anything at all. A stored procedure is immensely flexible and can be used for other things beyond just reporting or returning a value. It can add values to tables, build tables, or many other complex operations that can all be contained within a stored procedure. You can imagine that a stored procedure is in its most literal sense a chunk of SQL code that is called when it is, differing from the other two because it is capable of doing much more than just reporting.

**Summary**

We learned this week about how we can use views, functions and stored procedures in SQL. Ultimately, we want to use views in order to simplify complex SQL reporting and to add an abstraction layer to the database. Views, functions and stored procedures are all similar in that they all are a named set of SQL statements, but they have differences. Views can be used to compress complex select statements and allow for an abstraction layer, but functions can take in parameters and return single calculated values in addition to being able to return tables, while stored procedures don’t need to return anything and can be used with immense flexibility to accomplish many operations beyond select statements.