Danielle Shubat

February 24, 2021

Foundations of Databases & SQL Programming

Assignment 06

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

**Introduction**

In order to limit user access to the actual tables within a database and maintain data integrity, abstraction layers are used to provide buffers between database structure and user interface. Views, functions and stored procedures can create powerful abstraction layers and optimize database interface for the end user, allowing specifically selected data to be presented in a table-like format. In this paper I’ll discuss the similarities and key differences between view, functions and stored procedures, providing illustrative examples.

Explain when you would use a SQL View

Views are saved queries created in SQL through select statements that, when executed, present specifically selected data in a table-like format. Furthermore, data from one or more tables can be presented in one table using where and join statements, creating increasingly-specific table-like outputs. Data always remains within the table, not in the view. Views are commonly used to partition data, giving users access to certain field data based on their user type. A good example of partitioned data is an employee database that allows an HR Manager user to see sensitive employee information -such as Social Security Number- but restricts this information from being viewed by peers of the employee. In this case, multiple views from the same table(s) would be created and permissions granted respectively for the HR Manager and Employee user types.

Views are also a simple way to simplify the amount of information presented to a user. Instead of requiring an end user to run complex queries joining information from multiple tables, views can present this joined data easily. In Figure 1, we see an example of a view joining a Category and Product table that, when executed, returns data across the tables.

|  |  |
| --- | --- |
| Original Tables | View Showing CategoryName, ProductName and UnitPrice |
|  |  |

**Figure 1** A simple view showing categories, product names and unit price created by joining data from the category and product tables.

Views also provide a safeguard against user interface errors through Schemabinding. With access to the underlying tables, a user could mistakenly drop a table, even with a foreign key present. With Schemabinding, users can't drop a table, change or drop column names without an error message being presented. Once a view is bound to the underlying table, the view must visually mirror the table, protecting users from making any changes to a table that would break the view.

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

User defined functions are selectable code which can be used to return a single value. While they return information in much the same presentation as views, they are created with different syntax, and, unlike views, cannot return a null value. Both stored procedures and views can return null values. Most notably, functions use a different syntax than do views and require the .dbo database schema and Return statement, written with () in conclusion in order to run properly. Stored Procedures do not differ from views in terms of syntax except that they are Executed, rather than Selected, as highlighted in Figure 2.

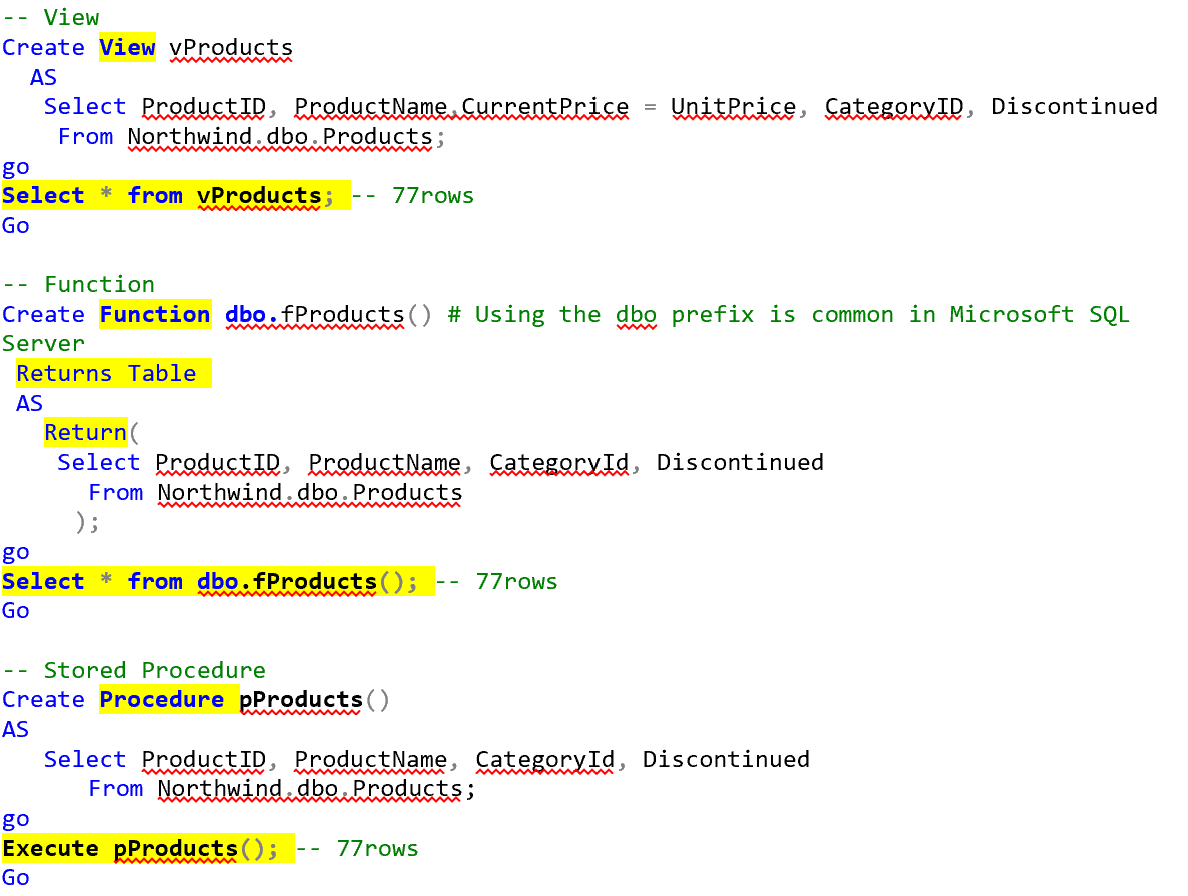


Figure 2 Code examples from Module 6 notes highlighting views, functions and procedures.

Summary

Views, functions and stored procedures are simple but vitally important means of creating abstraction layers between database data and users. While very similar in what they return and present, views, functions and stored procedures contain distinct differences that programmers must recognize in order to properly use these tools.