Name: NSchafi
Date: 8/15/2022
Course: IT FDN 130 A Su 22: Foundations Of Databases & SQL
GitHub URL: https://github.com/njschafi/DBFoundations

Assignment 6 – Views

Introduction

In this document, we will be exploring the use case for the VIEW statement. In addition to that, we will explore the difference and similarities between a VIEW, Function and Stored Procedure within SQL.

SQL View

The VIEW statement in SQL, put simply, is nothing more than a saved query definition. It does not hold actual data, only the definition of the view in the data dictionary. Using a view allows a user to pull data from multiple tables and display them as a saved query. The VIEW contains a set of predefined SQL queries to fetch data from the database. Example 1 shows the syntax for creating a view from a table:

```
Create View vPublicEmployeeInfo
As
 Select
   TitleOfCourtesy
  ,FirstName
  ,LastName
  ,Title
  From Northwind.dbo.Employees;
go
```

Example 1 - Creating a VIEW in SQL

The code above (in Example 1) shows how to create a VIEW. A user can name the VIEW however they please and select what columns they want displayed from one or more tables.

Most importantly, VIEW statements allow for security of data because they provide a "read" only display of the data and not the actual data. You can also limit what data can and can't be displayed to certain users. Different permissions can be granted on views and tables, so that you can show only a portion of data to a user. Example 2 shows how to apply permissions:

```
Use Module06Demos;
Deny Select On vPrivateEmployeeInfo to Public;
Grant Select On vPublicEmployeeInfo to Public;
```

Example 2 - Applying permissions on VIEWS

The permissions applied above (in Example 2) would make the "vPrivateEmployeeInfo" View not displayable to the group named "Public". Whereas, the View "vPublicEmployeeInfo" is displayable to the group named "Public".

Views, Functions and Stored Procedures

Similar to many other programming languages, SQL provides users with a way of creating functions or "User Defined Functions". There are two basic function types in SQL: one that returns a table of values and one that returns a single value (Scalar Function). A function can be used to return/display table specific table values similar to a VIEW statement, see Example 3 below:

```
Create Function dbo.fProducts()
 Returns Table
 AS
   Return(
    Select ProductID, ProductName, CategoryId, Discontinued
        From Northwind.dbo.Products
       );
```

Example 3 - Using a function to display data

In Example 3 above, the function returns a table with the selected columns to the user. This is very similar to using a VIEW or even a SELECT statement. Where a function differs greatly from a view is the capability to pass in parameters, resulting in a change to a result-set of a query. Example 4 below showcases a function with parameters:

```
Alter Function dbo.fProducts(@CategoryId int)
 Returns Table
 AS
   Return(
   Select ProductID, ProductName
    From Northwind.dbo.Products
        Where CategoryID = @CategoryId
   );
go
Select * From dbo.fProducts(1);
go
```

Example 4 – Passing in a parameter to a function

In Example 4 above, the parameter being passed in is the "@CategoryID" defined as an integer type. If a user passes anything other than an integer, the function will not work and throw an error. Further down, you can see the code calling the function with the parameter of 1. This will return results only with the CategoryID equal to 1.

Another useful tool within SQL are Stored Procedures (Sprocs or Procs), which are a named set of SQL statements. A Stored Procedure is prepared SQL code that is saved so it can be reused over and over again. This is similar to the concept of VIEWS and Functions where all 3 (Stored Procedures, VIEWs, Functions) can display data to the user by just being called. Example 5 shows the syntax to create a Standard Procedure:

```
Create Procedure pProducts()
AS
   Select ProductID, ProductName, CategoryId, Discontinued
        From Northwind.dbo.Products;
go
Execute pProducts(); -- 77rows
Go
```

Example 5 - Using a Stored Procedure to display data

The code above, in Example 5, creates the Stored Procedure called “pProducts()”. When executed later on in the code, it will display the selected data from the database table “Products”. It differs from a VIEW since a Stored Procedure can take in parameters and it differs from a Function because it does not have to return anything (it just executes).

Summary

In conclusion, Module 6 introduces the concepts of the VIEW statement, Functions and Stored Procedures in SQL. All enable the user to display database data in various ways (rather than using the SELECT statement) and each performs the task differently. To quickly recap their differences: A VIEW is a saved query that user can call upon at any time, a Function can return a table based on parameters (optional), and Stored Procedure can execute a saved set of SQL statements.