Name: Derek Benak
Date: November 20, 2023
Course: Foundations of Databases & SQL Programming
Assignment 6
https://github.com/DerekJB1987/DBFoundations.git

Views, Functions, and Stored Procedures

Introduction

This week I started learning about Views, Functions, and Stored Procedures that are used when a SQL programmer wants to save, show, or enhance more complex SQL statements for repeated use in the future rather than recreating the wheel each time.

When to use Views

A programmer / data analyst would want to use a SQL **View** for reporting purposes to show the SQL code and/or retrieved results from the tables in the database. This is often the case when the programmer wants to protect the private data with permissions and only allow access to an employee or customer data through appropriate views.

```
CREATE VIEW vEmployees
WITH SCHEMABINDING
    AS
        SELECT
[EmployeeID],[EmployeeFirstName],[EmployeeLastName],[ManagerID]
        FROM dbo.Employees;


CREATE VIEW vInventories
WITH SCHEMABINDING
    AS
        SELECT [InventoryID],[InventoryDate],[EmployeeID],[ProductID],[Count]
        FROM dbo.Inventories;
```

Figure 1: Example of SQL code where a user is saving multiple individual tables (Employees, and Inventories) as separate views that can be used to extract data for reporting purposes

```
DENY SELECT ON Employees TO PUBLIC;
DENY SELECT ON Inventories TO PUBLIC;
```

```
GRANT SELECT ON vEmployees TO PUBLIC;
GRANT SELECT ON vInventories TO PUBLIC;
```

Figure 2: Example of SQL code where a user is setting permissions to Deny access on the primary tables and Granting access to the views for the same tables

What are differences and similarities between a View, Function and Stored Procedure

SQL Views, Functions, and Store Procedures are tools in SQL used for selecting report data in a database.

A **View** is a virtual table based on a Select statement used to retrieve a result-set of data from tables in a database.

```
CREATE VIEW vCategories
WITH SCHEMABINDING
    AS
        SELECT [CategoryID],[CategoryName]
            FROM dbo.Categories;
```

Figure 3: Example of SQL code where a user is creating a "Base" view to show data from the Categories table

There are two types of **Functions** in SQL server, functions that return a table of values and functions that return a single value. There are built-in functions, and a user can create custom functions. Functions and views are similarly named and are used to return a desired set of results

```
-- View
Create View vProducts
  AS
   Select ProductID, ProductName,CurrentPrice = UnitPrice, CategoryID, Discontinued
    From Northwind.dbo.Products;
go
Select * from vProducts; -- 77rows
Go

-- Function
Create Function dbo.fProducts() # Using the dbo prefix is common in Microsoft SQL
Server
 Returns Table
 AS
   Return(
    Select ProductID, ProductName, CategoryId, Discontinued
        From Northwind.dbo.Products
       );
go
Select * from dbo.fProducts(); -- 77rows
go
```

Figure 4: Example of SQL code showing a View and custom function named fProducts() to return a desired set of results

A function can use parameters to change the results of the query.

```
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); -- 12 rows
go
```

Figure 5: Example of SQL code where a user is using a custom function with a parameter to refine the results that will be retrieved

A **Stored Procedure** is a named set of SQL statements that is saved and can be reused repeatedly. You can also pass parameters to a stored procedure like a function.

Summary

This writeup highlights the fundamentals of the tools available in SQL to save more complex SQL statements for repeated use in the future and to retrieve results from the database for reporting purposes.