Greg Parker
Aug 6th, 2024
IT FDN 130 A Su 24: Foundations Of Databases & SQL Programming
Assignment 06

VIEW

Introduction

This week's assignment built off last week's topic of JOIN statements with the introduction of VIEWS. In this week's writeup I will explain when you would use a SQL VIEW as well as explain what the differences and similarities between a View, Function, and Stored Procedure.

When to VIEW

Views are useful for a variety of reasons when building a query. When used, views simplify otherwise complex code by allowing you to name the view and then recall it without needing to rewrite (reusability!) the code. Views also allow for addition of user security settings by preventing users from accessing/viewing specific tables.

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

Example of security – allowing access to select from the view not the table[1]

Lastly, views help with data abstraction in that they allow you to change the view of the data without changing the underlying tables that house the data.

View/Function/Stored Procedure

A view is a virtual table based on the result set of a SQL query. It doesn't store data itself but displays data stored in other tables. The best time to use a view is when simplifying complex queries, enhancing security by restricting access to specific data, and providing a consistent, abstracted interface to the data.

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

Example of a basic view[2]

A function is an object that performs a calculation or operation and returns a single value or a table. The best time to use a function is when capturing a logic statement to be reused later, performing calculations, and returning specific data. Functions should not be used with performing actions that modify the database like inserting or updating.

```
CREATE FUNCTION GetTotalSales (@CustomerID INT)
RETURNS DECIMAL(10, 2)
AS
BEGIN
    DECLARE @Total DECIMAL(10, 2);
    SELECT @Total = SUM(Amount)
    FROM Sales
    WHERE CustomerID = @CustomerID;
    RETURN @Total;
END;
```

Example of a Function[3]

A Stored Procedure is a one or more SQL statements that can perform complex operations including database modifications. These procedures are best used to capture business logic, automating repetitive tasks, and improving performance. One of the main drawbacks to stored procedures is that they can be too complex and become difficult to manage and debug. Views and Functions don't have the same problem.

```
CREATE PROCEDURE UpdateOrderStatus
@OrderID INT,
@Status VARCHAR(20)
AS
BEGIN
    UPDATE Orders
    SET Status = @Status
    WHERE OrderID = @OrderID;
END;
```

Example of a Store Procedure to update the DB whenever there is a new Order Number[4]

Summary

To summarize this week, I learned a lot Views. I learned that views are the best way to recall complex queries so they can be used over again as well as a way you protect your tables from adding in layers of security. Stored procedures are great for automation and functions are great tools for running calculations.

Bibliography

1 Author: RRoot – Module6Notes.docx

2 Author: RRoot – Module6Notes.docx

3 Author: CoPilot – bing.com

4 Author: CoPilot – bing.com