Name: Mitsuyo Kawano

Date: Feb 22, 2023

Course: Foundations of Databases & SQL Programming

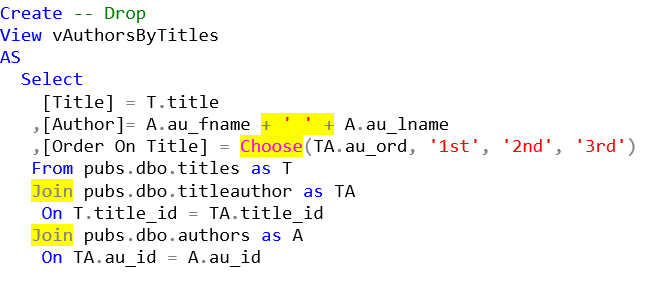
Assignment 06

<https://github.com/mitsuyojp/DBFoundations>

**Assignment 6 – View, Function, and Stored Procedure**

Introduction  
In this module, we learned about View, Function and Stored Procedure.

As our SQL Select statements become complex, we may decide to save them in a text file for repeated use. Alternatively, we can save our Select statements within a database's file as a SQL View, Function, or Stored Procedure.

1. Explain when you would use a SQL View.   
   We learned about View previous module that we can use View to store complex Select statement. We can run saved code using only a simple Select statement which makes complex SQL code much easier to use. Here is the example.

Once we created this View, all we need is the following statement to run the code.  
  
 Select **\* from** vAuthorsByTitles;

There are two types of Views: Reporting View and Base View.

* **Reporting View**Any view that is used to extract data for reporting purposes is called a "Reporting View." They can save simple or complex Select statements, but more complex ones are typical.
* **Base View**In addition to any reporting views, we createeach table in a database should have a "Base" or "Basic" view to show data from that table. When we make a table, we create a **base view** and then **restrict access to the table** while **allowing access to the View**. Base views allow people to use your data in the "Abstract."

In this module, we learned that View is not only storing complex select statements, but it also restricts the access to the data, and also protect the data from unnecessary changes. So here is the conclusion.  
  
When would you use a SQL View?

* + When we want to store complex select statement to save time for future uses, or for other people to run the code easily with a simple select statement.
  + Structure data in a way that users or classes of users find natural or intuitive.
  + Restrict access to the data in such a way that a user can see and (sometimes) modify exactly what they need and no more.
  + Summarize data from various tables which can be used to generate reports.

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

In addition to SQL Server's built-in functions, we can create custom functions. These are often called User Defined Functions or just UDFs. There are two basic types of functions: functions that return a table of values and functions that return a single value.

Functions and Views are similar. Here are two examples that show how they are similar:

**-- 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

* Functions with Parameters  
  Unlike views, Functions can use parameters to change the result of the query as it is executed like this.

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

* Scalars Functions   
  Unlike Views, we can create UDFs to return a single(scalar) Value as an expression. Unlike parameters in table functions, parameters in scalar functions are very useful!

Create Function **dbo**.**MultiplyValues**(@Value1 Float, @Value2 Float)

Returns Float

As

Begin

Return(Select @Value1 \* @Value2);

End

go

-- Calling the function

Select Tempdb.**dbo**.**MultiplyValues**(4, 5);

go

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

* Stored Procedures  
   Like Views or Functions, Stored Procedures (Sprocs or Procs) are a Named Set of SQL Statements. Creating Views, Functions, and Stored Procedures are all similar. Here is an example of Stored Procedures:

**-- Stored Procedure**

Create **Procedure pProducts**()

AS

Select ProductID, ProductName, CategoryId, Discontinued

From Northwind.dbo.Products;

go

**Execute pProducts();** -- 77rows

Go

To run the code of Stored Procedures, we have to use an Execute statement. (Not Select statement like View and Functions). Also with Stored Procedures, we can store not only Select statements, but also other statements such as Create etc.

Summary

Explain when you would use a SQL View.

* + When we want to store complex select statement to save time for future uses, or for other people to run the code easily with a simple select statement.
  + Structure data in a way that users or classes of users find natural or intuitive.
  + Restrict access to the data in such a way that a user can see and (sometimes) modify exactly what they need and no more.
  + Summarize data from various tables which can be used to generate reports.

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

* + View, Function and Stored Procedure are very similar, they can store Select Statements and run the stored code with very simple statement.
  + Unlike views, Functions can use parameters to change the result of the query.
  + Scalars Functions is unlike Views, we can create UDFs to return a single(scalar) Value as an expression.
  + To run the code of Stored Procedure, we have to use Execute statement. (Not Select statement like View and Functions).
  + View and Functions can store only Select statements. With Stored Procedures, we can store not only Select statements, but also other statements such as Create etc.