### Michele Murphy

August 16, 2020

IT Foundations of Database Management

Assignment 06

[GitHub - mmurph88UW](https://github.com/mmurph88UW/DBFoundations)

# Views, Functions & Stored Procedures

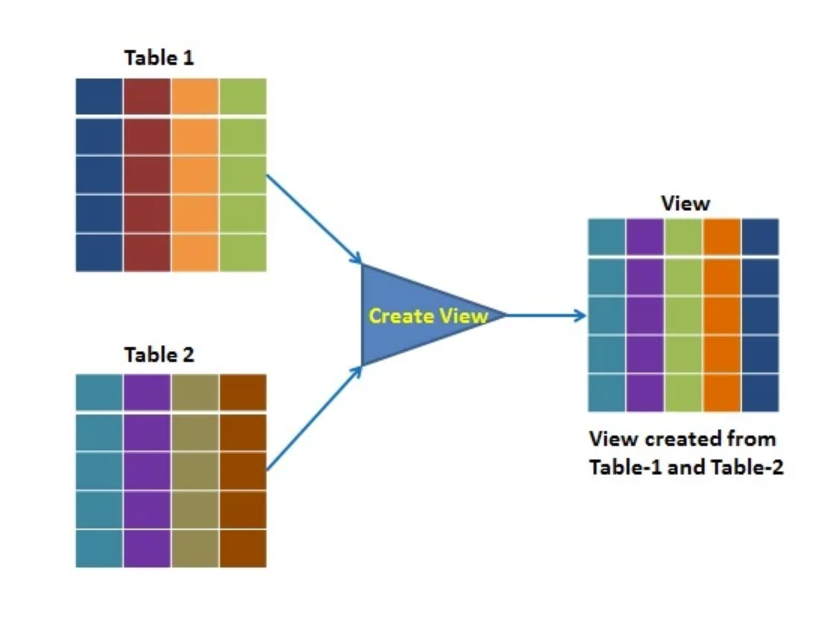
## Introduction

In this assignment we will look at when you would use a SQL View and explain what the differences and similarities are between a View, Function, and Stored Procedure.

## When would you use a SQL View?

SQL Views are used for inserting an abstraction layer between the user and the actual base tables, enabling a way to customize or streamline a user’s experience of how data is accessed or presented that is accessed through the View. SQL Views also allow for an additional layer of database security to exist as user permissions to the base tables can be removed and the user can be granted access to the SQL Views instead.

SQL Views are saved select statements that act as a type of virtual table in that they allow users to access data in ways that are similar to access the underlying base tables. (Figure 1.1 <https://www.datacamp.com/community/tutorials/views-in-sql> )(external site)



***Figure 1.1: Graphic Representation for Creating a SQL View***

A SQL View is one of the ways that SQL Code or Select statement can be stored in the database. Additional change management protections such as Schemabinding can be added to a View so that unintentional mistakes are not made when changes to the actual tables are attempted that could otherwise orphan a View or a Column within a View.

## The Differences and Similarities between Views, Functions, and Stored Procedures

The main thing that SQL Views, Functions and Stored Procedures have in common is that all 3 allow SQL Code to be saved in a database. SQL Views and Functions are named Select statements that are stored in the database while a Stored Procedure is also a way to store SQL code in the database but is not restricted to only containing Select statements.

### Additional differences between Views, Functions and Stored Procedures are as follows:

* A View or a Function is a Select Statement stored in the database tables.
* A Stored Procedure is also stored in the database but is not imited to just SELECT Statements and contain other DML operations such as: INSERT, UPDATE, DELETE
* You do not use parenthesis when using a View but as Functions and Stored Procedures can have Parameters associated with them, typically you will need to use a parenthesis when calling a Function or Executing a Stored Procedure.
* Unlike a view, a Function can return an individual value such as an integer, a number or date or a single word

### Running a VIEW or Function and executing a Stored Procedure

* A View or Function is run in a SELECT Statement or used in a FROM Clause whereas a Stored Procedure is group of statements that can be Executed.
* You SELECT from a View

SELECT \* FROM vProducts;

* You also SELECT from Function

SELECT \* FROM fProducts;

* You do not SELECT from a Stored Procedure.
  + Instead you 'Run its Code' with the EXECUTE Command

Execute pProducts();

### CREATE Statements for Views, Functions and Stored Procedures

The CREATE Statements for Views, Functions and Stored Procedures all have the 'AS' keyword in the create statement but there are additional differences each type of CREATE statement has.

An example of a CREATE Statement for SQL Views is shown below. (Figure 2.1)

GO

CREATE

VIEW vCustomersByLocation

AS

SELECT CompanyName AS CustomerName

,City

,IsNUll(Region,Country) AS Region

,Country

FROM dbo.Customers;

GO

SELECT \* FROM dbo.vCustomersByLocation;

GO

***Figure 2.1: SQL View CREATE Statement***

Function CREATE Statements differ from Views and Stored Procedure Statements as you must have the

* 'RETURNS TABLE' in the CREATE Statement as
* RETURNS TABLE at beginning
* RETURN after AS
* SELECT Statement portion of CREATE Statement must be encapsulated in Parenthesis
* You must use namespace (typically dbo.) preface in CREATE Statement

An example of a CREATE Statement for a SQL Function is shown below. (Figure 2.2)

GO

CREATE

FUNCTION dbo.fCustomersByLocation()

RETURNS TABLE

AS

RETURN (

SELECT CompanyName AS CustomerName

,City

,IsNUll(Region,Country) AS Region

,Country

FROM dbo.Customers AS c);

GO

SELECT \* FROM dbo.fCustomersByLocation();

GO

***Figure 2.2: SQL Function CREATE Statement***

An example of a CREATE Statement for a SQL Stored Procedure is shown below. (Figure 2.3)

GO

CREATE

PROCEDURE pCustomersByLocation

AS

SELECT CompanyName AS CustomerName

,City

,IsNUll(Region,Country) AS Region

,Country

FROM dbo.Customers;

GO

EXEC pCustomersByLocation;

GO

***Figure 2.3: SQL Stored Procedure CREATE Statement***

## Summary

In summary, we have just reviewed that SQL Views are saved select statements that act as a type of virtual table in that they allow users to access data in ways that are similar to access the underlying base tables. SQL Views are used as an abstraction layer and enable a way to customize a user’s experience in terms of how how data is accessed in a database. SQL Views also bring additional security in that you can adjust user permissions to only be able to access the Views vs the base tables. Schemabinding can also be put into place to ensure tables changes a not made that would orphan and View or a Column within a View.

We also took a look at some of the differences and similarities are between a View, Function, and Stored Procedure. A SQL View or Function is a Select Statement stored in the database while a Stored Procedure can also be a Select Statement stored in the database, a Stored Procedure is not limited to only being a Select Statement. A Stored procedure can include the oter DML operations such as INSERT, UPDATE and DELETE Statements. In addition, we also review the differeces View, Function, and Stored Procedure have in the CREATE Statements and in how each are Run or executed when accessed.