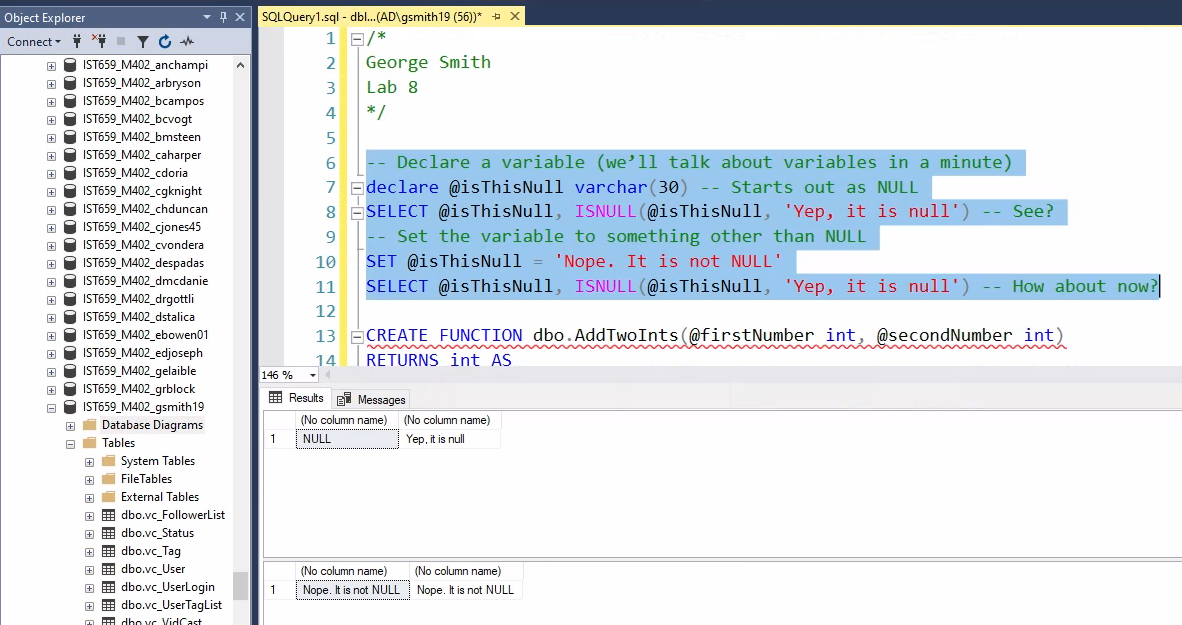
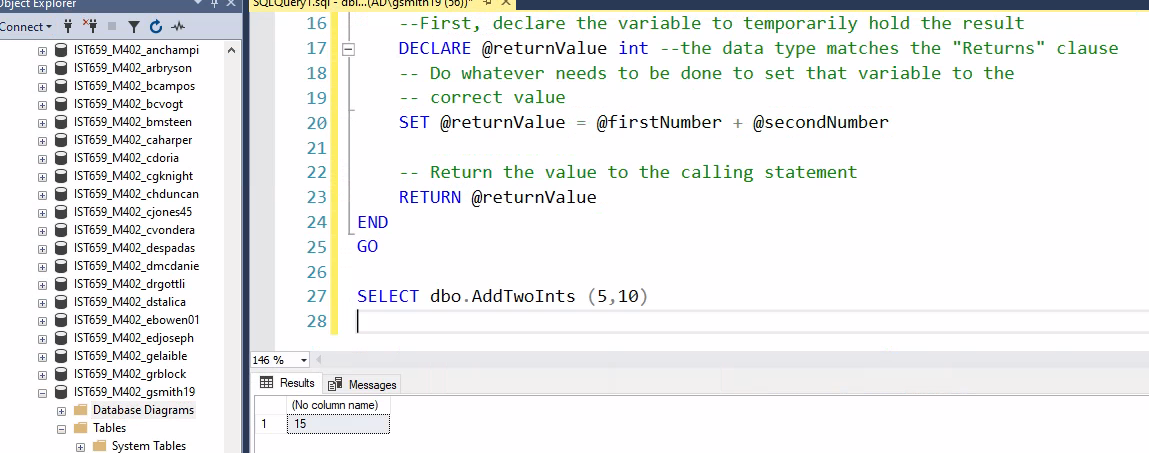
George Smith

Lab 8

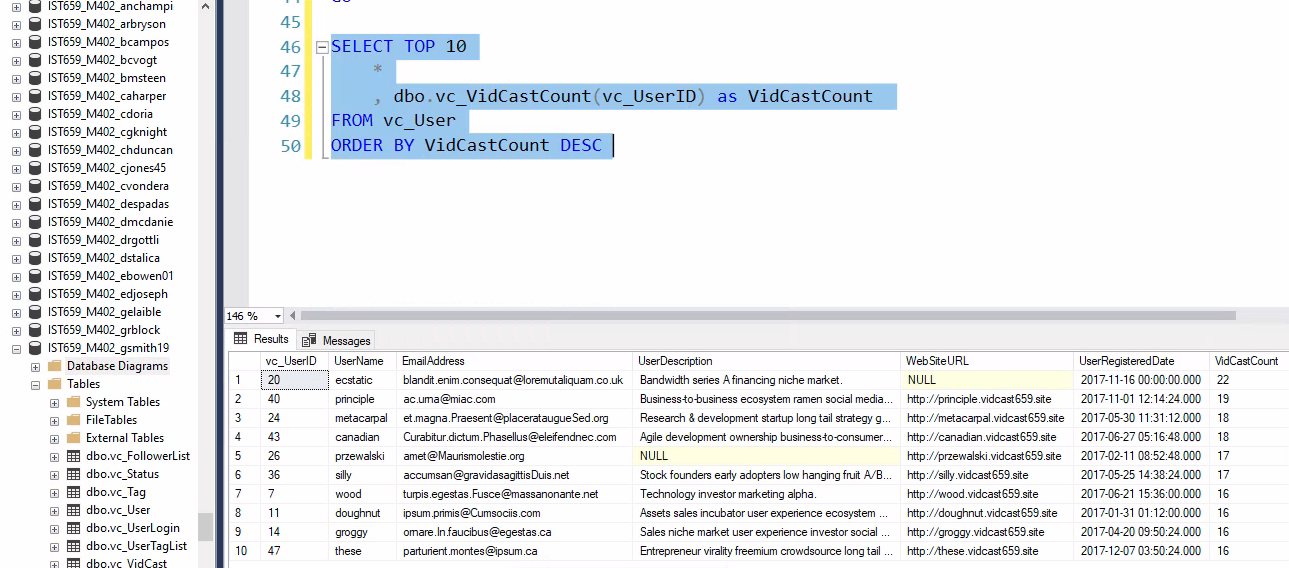
1.



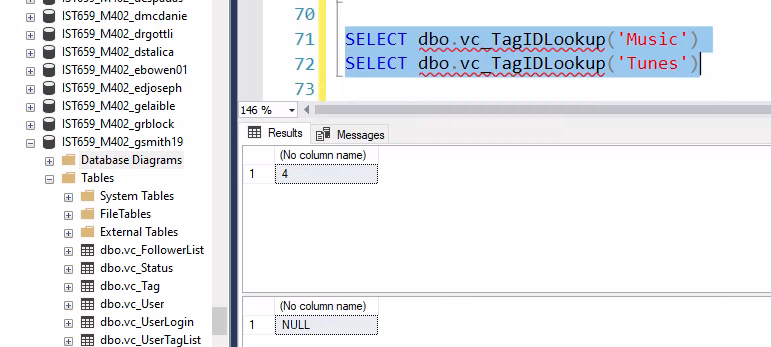
2.



3.

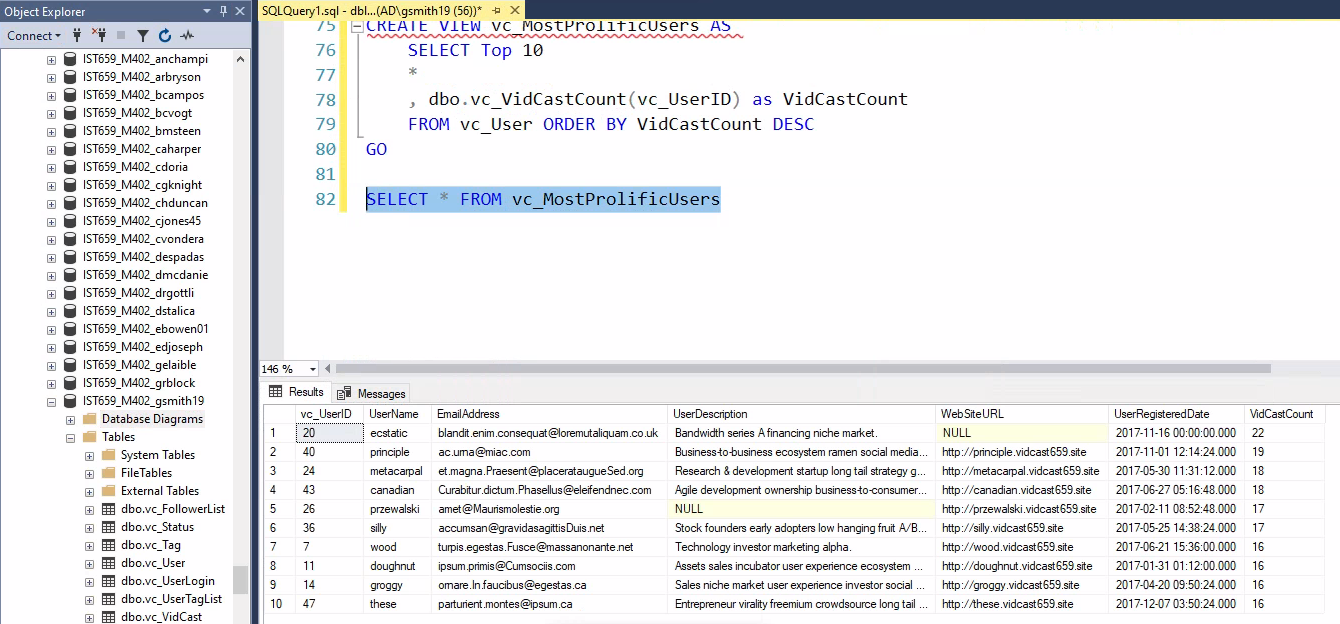


4.

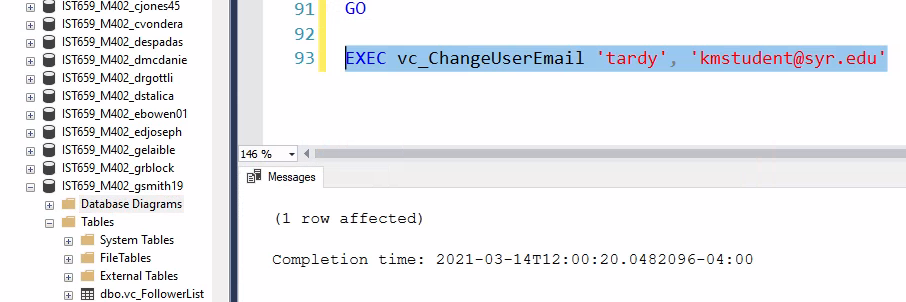


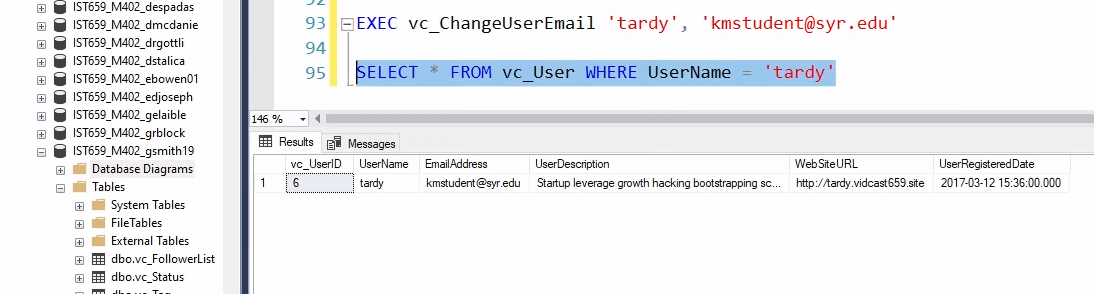
5. The above code is returning the amount of TagIDs where the TagText has either the parameter music of tunes. The null value shows that the parameter Tunes did not have any results.

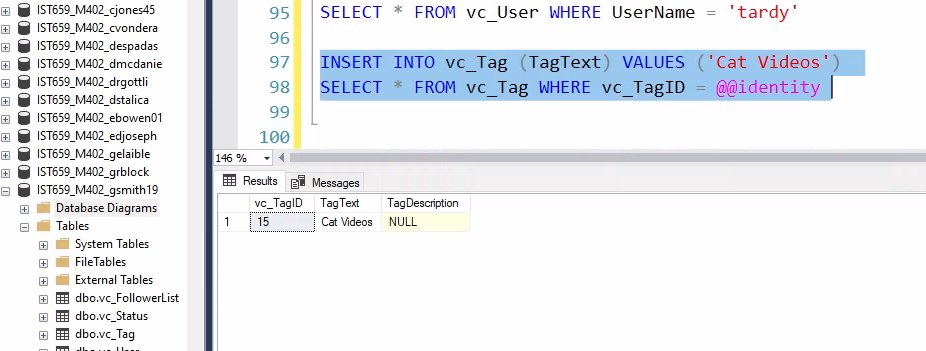
6.



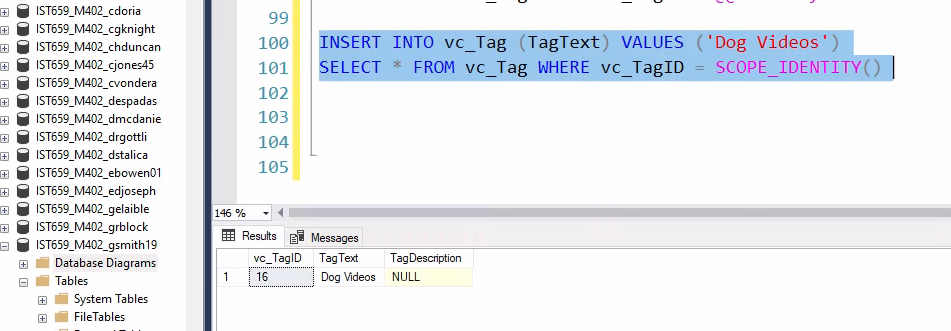
7.



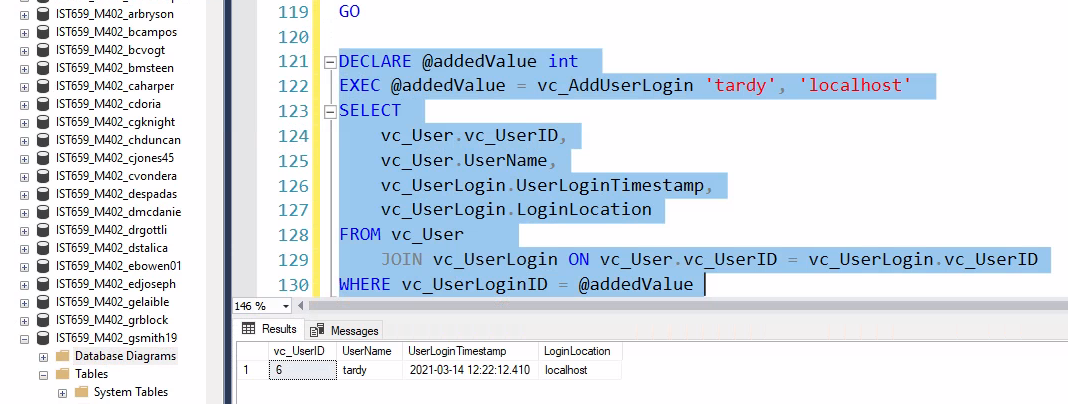
8. 

9. 

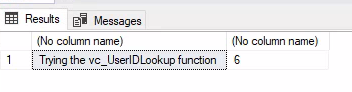
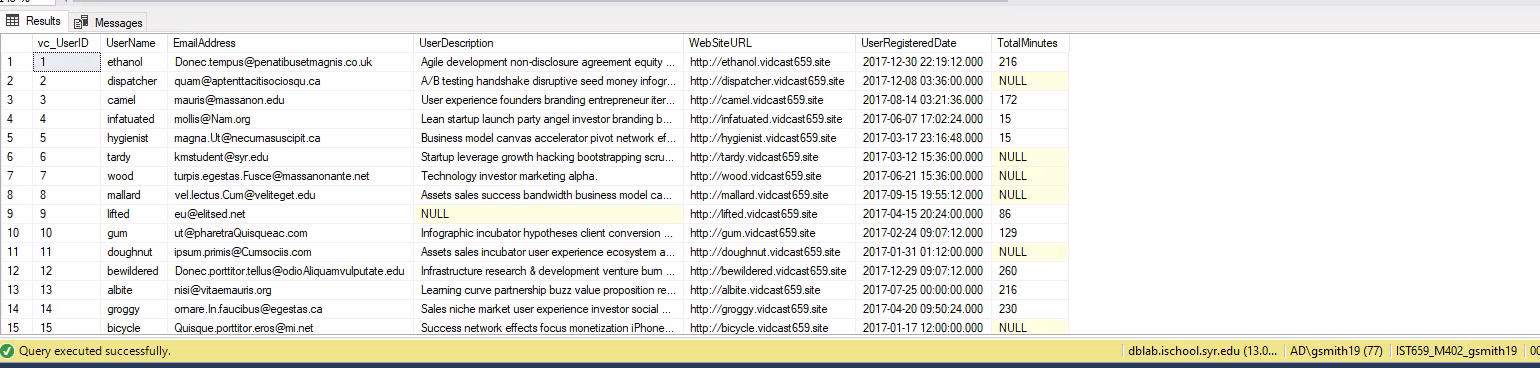
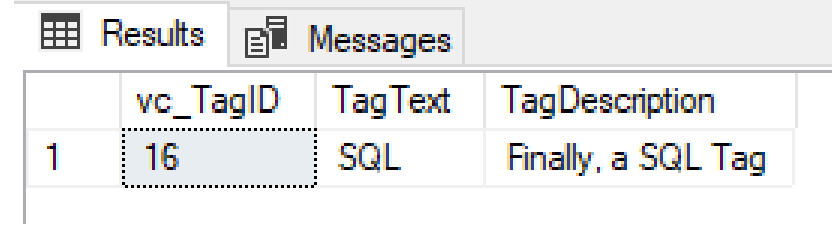
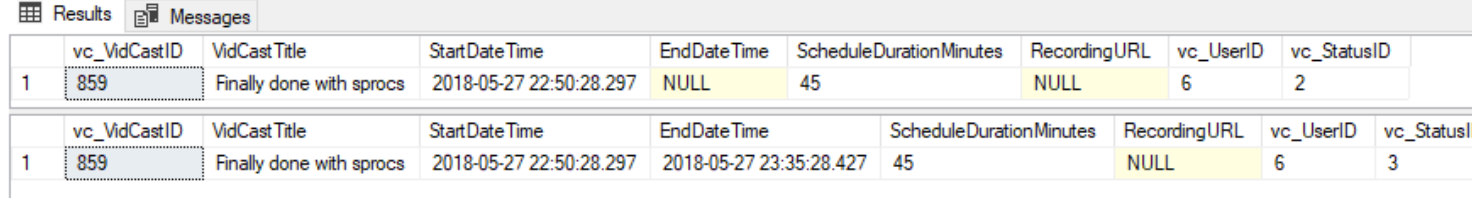
10.



11.



**Part 2**

1. 
2. 
3. 
4. 
5. 
6. 

**SQL CODE:**

**/\***

**George Smith**

**Lab 8**

**\*/**

**-- Declare a variable (we’ll talk about variables in a minute)**

**declare @isThisNull varchar(30) -- Starts out as NULL**

**SELECT @isThisNull, ISNULL(@isThisNull, 'Yep, it is null') -- See?**

**-- Set the variable to something other than NULL**

**SET @isThisNull = 'Nope. It is not NULL'**

**SELECT @isThisNull, ISNULL(@isThisNull, 'Yep, it is null') -- How about now?**

**CREATE FUNCTION dbo.AddTwoInts(@firstNumber int, @secondNumber int)**

**RETURNS int AS**

**BEGIN**

**--First, declare the variable to temporarily hold the result**

**DECLARE @returnValue int --the data type matches the "Returns" clause**

**-- Do whatever needs to be done to set that variable to the**

**-- correct value**

**SET @returnValue = @firstNumber + @secondNumber**

**-- Return the value to the calling statement**

**RETURN @returnValue**

**END**

**GO**

**SELECT dbo.AddTwoInts (5,10)**

**-- Function to count the Vidcasts made by a given User**

**CREATE FUNCTION dbo.vc\_VidCastCount (@userID int)**

**RETURNS int AS -- COUNT () is an integer value, so return it as an int**

**BEGIN**

**DECLARE @returnValue int -- matches the function's return type**

**/\***

**Get the count of the VidCasts for the provided userID and assign that value to @returnValue. Note that we use the @userID parameter in the WHERE calue to limit our count to that User's Vidcast records.**

**\*/**

**SELECT @returnValue = COUNT (vc\_UserID) FROM vc\_VidCast**

**WHERE vc\_VidCast.vc\_UserID = @userID**

**-- Return @returnValue to the calling code.**

**RETURN @returnValue**

**END**

**GO**

**SELECT TOP 10**

**\***

**, dbo.vc\_VidCastCount(vc\_UserID) as VidCastCount**

**FROM vc\_User**

**ORDER BY VidCastCount DESC**

**GO**

**--Function to retrieve the vc\_TagID for a given tag's text**

**CREATE FUNCTION dbo.vc\_TagIDLookup(@tagText varchar (20))**

**RETURNS int AS -- vc\_TagID is an int, so we'll match that**

**BEGIN**

**DECLARE @returnvalue int --Matches the funtion's return type**

**/\***

**Get the vc\_TagID of the vc\_Tag record whose TagText matches the parameter and assign that value to @returnValue.**

**\*/**

**SELECT @returnValue = vc\_TagID FROM vc\_Tag**

**Where TagText = @tagText**

**--Send the vc\_TagID back to the caller**

**RETURN @returnValue**

**END**

**SELECT dbo.vc\_TagIDLookup('Music')**

**SELECT dbo.vc\_TagIDLookup('Tunes')**

**-- Create a view to retrieve the top 10 vc\_Users and their VidCast counts**

**CREATE VIEW vc\_MostProlificUsers AS**

**SELECT Top 10**

**\***

**, dbo.vc\_VidCastCount(vc\_UserID) as VidCastCount**

**FROM vc\_User ORDER BY VidCastCount DESC**

**GO**

**SELECT \* FROM vc\_MostProlificUsers**

**--create a procedure**

**CREATE PROCEDURE vc\_ChangeUserEmail (@userName varchar (20), @newEmail varchar(50))**

**AS**

**BEGIN**

**UPDATE vc\_User Set EmailAddress = @newEmail**

**Where UserName = @userName**

**END**

**GO**

**EXEC vc\_ChangeUserEmail 'tardy', 'kmstudent@syr.edu'**

**SELECT \* FROM vc\_User WHERE UserName = 'tardy'**

**INSERT INTO vc\_Tag (TagText) VALUES ('Cat Videos')**

**SELECT \* FROM vc\_Tag WHERE vc\_TagID = @@identity**

**INSERT INTO vc\_Tag (TagText) VALUES ('Dog Videos')**

**SELECT \* FROM vc\_Tag WHERE vc\_TagID = SCOPE\_IDENTITY()**

**-- New Procedure**

**CREATE OR ALTER PROCEDURE vc\_AddUserLogin (@userName varchar(20), @loginFrom varchar(50))**

**AS**

**BEGIN**

**DECLARE @userID int**

**SELECT @userID = vc\_UserID FROM vc\_User**

**WHERE UserName = @userName**

**INSERT INTO vc\_UserLogin (vc\_UserID, LoginLocation)**

**VALUES (@userID, @loginFrom)**

**RETURN SCOPE\_IDENTITY()**

**END**

**GO**

**DECLARE @addedValue int**

**EXEC @addedValue = vc\_AddUserLogin 'tardy', 'localhost'**

**SELECT**

**vc\_User.vc\_UserID,**

**vc\_User.UserName,**

**vc\_UserLogin.UserLoginTimestamp,**

**vc\_UserLogin.LoginLocation**

**FROM vc\_User**

**JOIN vc\_UserLogin ON vc\_User.vc\_UserID = vc\_UserLogin.vc\_UserID**

**WHERE vc\_UserLoginID = @addedValue**

**-- Part 2**

**-- create a function to retrieve a vc\_UserID for a given user name**

**CREATE FUNCTION dbo.vc\_UserIDLookup(@userName varchar(20))**

**RETURNS int AS**

**BEGIN**

**DECLARE @returnValue int**

**SELECT @returnValue = (vc\_UserID) FROM vc\_User**

**WHERE Username = @userName**

**RETURN @returnValue**

**END**

**GO**

**SELECT 'Trying the vc\_UserIDLookup function' , dbo.vc\_UserIDLookup('tardy')**

**-- statement**

**CREATE FUNCTION dbo.vc\_TagVidCastCount (@VidCastID int)**

**RETURNS int AS**

**BEGIN**

**DECLARE @returnvalue int**

**SELECT @returnValue = COUNT (vc\_VidCastID) FROM vc\_VidCast**

**WHERE vc\_Vidcast.vc\_VidCastID = @VidCastID**

**RETURN @VidCastID**

**END**

**GO**

**SELECT**

**vc\_Tag.TagText,**

**dbo.vc\_TagVidCastCount(vc\_Tag.vc\_TagID) as VidCasts**

**FROM vc\_tag**

**--Question 3**

**CREATE FUNCTION dbo.vc\_VidCastDuration(@VidCastDuration int)**

**RETURNS int AS**

**BEGIN**

**DECLARE @returnValue int**

**SELECT @returnValue = DATEDIFF(n, StartDateTime, EndDateTime) FROM vc\_VidCast**

**WHERE vc\_VidCast.vc\_UserID = @VidCastDuration**

**RETURN @returnValue**

**END**

**GO**

**SELECT**

**\***

**, dbo.vc\_VidCastDuration(vc\_UserID) as TotalMinutes**

**FROM vc\_User**

**-- Create view**

**CREATE VIEW vc\_TagReport AS**

**SELECT vc\_Tag.TagText,**

**dbo.vc\_TagVidCastCount(vc\_Tag.vc\_TagID) as VidCasts**

**FROM vc\_Tag**

**ORDER BY VidCasts DESC**

**GO**

**SELECT \* FROM vc\_TagReport**

**-- Create a stored procedure**

**CREATE PROCEDURE vc\_AddTag(@tagText varchar(20), @description varchar(100)=NULL)**

**AS**

**BEGIN**

**DECLARE @vc\_TagID int**

**SELECT @vc\_TagID = vc\_TagID FROM vc\_Tag**

**Where TagText = @tagText**

**INSERT INTO vc\_Tag (vc\_TagID, TagDescription)**

**VALUES (@vc\_TagID, @description)**

**RETURN SCOPE\_IDENTITY()**

**END**

**GO**

**DECLARE @newTagID int**

**EXEC @newTagID = vc\_AddTag 'SQL', 'Finally, a SQL Tag'**

**SELECT \* FROM vc\_Tag where vc\_TagID = @newTagID**

**CREATE PROCEDURE vc\_FinishVidCast (@VidCastID int, @finished int )**

**AS**

**BEGIN**

**UPDATE vc\_VidCast**

**set vc\_StatusID=@finished**

**where vc\_VidCastID = @VidCastID**

**end**

**go**

**DECLARE @newVC int**

**INSERT INTO vc\_VidCast**

**(VidCastTitle, StartDateTime, ScheduleDurationMinutes, vc\_UserID,**

**vc\_StatusID)**

**VALUES (**

**'Finally done with sprocs'**

**, DATEADD(n, -45, GETDATE())**

**, 45**

**, (SELECT vc\_UserID FROM vc\_User WHERE UserName = 'tardy')**

**, (SELECT vc\_StatusID FROM vc\_Status WHERE StatusText='Started')**

**)**

**SET @newVC = SCOPE\_IDENTITY()**

**SELECT \* FROM vc\_VidCast WHERE vc\_VidCastID = @newVC**

**EXEC vc\_FinishVidCast @newVC**

**SELECT \* FROM vc\_VidCast WHERE vc\_VidCastID = @newVC**