Sana Khan

Lab 8

In your own words, in your answers document, describe what lines 49 through 53 above actually do. Also, how is it that this code knows that the vc\_User record with vc\_UserID = 20 has 22 vc\_VidCast records?

This code is selecting the top 10 results which is using a function to count the number of videos a user has made. It’s able to count the number of videos because it’s counting the number of times the user id appears and showing that value

In your own words, in your answers document, describe what lines 75 and 76 above actually do. Also, when line 76 executed, we received a NULL from SQL Server. How come?

This code is showing the TagID for the corresponding tag that we are trying to look up. The second line came back as NULL because that tag does not exist in the database

In your own words, in your answers document, describe what lines 79 through 87 above are doing.

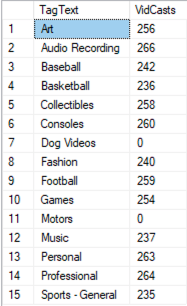
This is creating a view based on the function we previously created

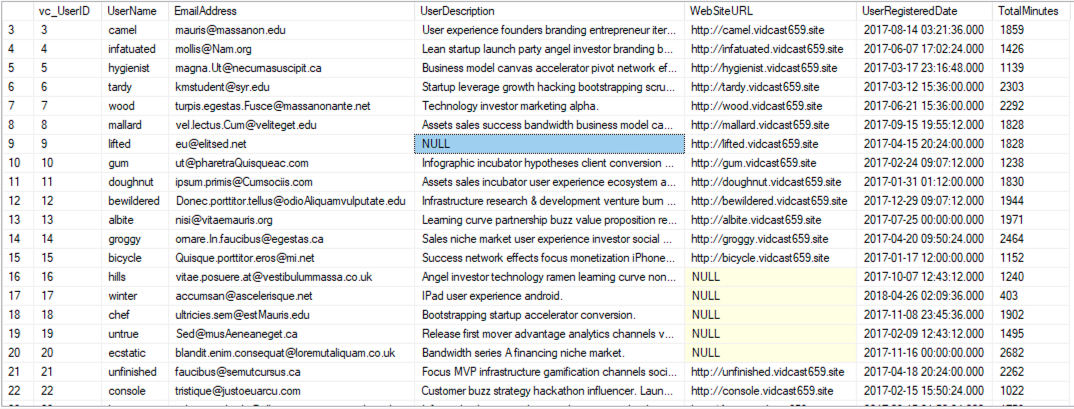
In your own words, in your answers document, describe what lines 91 through 104 above are doing.

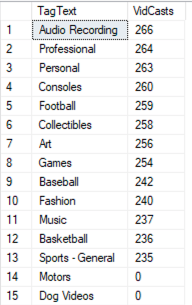
This stored procedure is updating the email address for a specified user

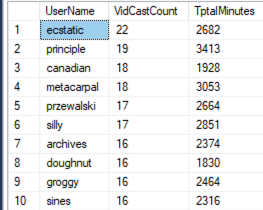
Your UserLoginTimestamp value will be different than the one shown. On your answers doc, explain why this is.

The UserLoginTimestamp is different than the lab because this value is based on when the stored procedure is ran. Because











I wasn’t able to get the last procedure to work so I do not have a screenshot for it

-- 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

DECLARE @returnValue int

SET @returnValue = @firstNumber + @secondNumber

RETURN @returnValue

END

GO

SELECT dbo.AddTwoInts(5,10)

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

RETURNS int AS

BEGIN

DECLARE @returnValue int

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

WHERE vc\_VidCast.vc\_UserID = @userID

RETURN @returnValue

END

GO

SELECT TOP 10

\*

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

FROM vc\_User

ORDER BY VidCastCount DESC

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

RETURNS int AS

BEGIN

DECLARE @returnValue int

SELECT @returnValue = vc\_TagID FROM vc\_Tag

WHERE TagText = @tagText

RETURN @returnValue END

GO

SELECT dbo.vc\_TagIDLookup('Music')

SELECT dbo.vc\_TagIDLookup('Tunes')

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 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 ('Dog Videos')

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

GO

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

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

RETURNS int AS

BEGIN

DECLARE @returnValue int

SELECT @returnValue = vc\_UserID FROM uc\_User

WHERE UserName = @userNAME

RETURN @returnValue

END

GO

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

CREATE FUNCTION dbo.vc\_TagVidCastCount(@tagID int)

RETURNS int AS

BEGIN

DECLARE @returnValue int

SELECT @returnValue = COUNT(vc\_TagID) FROM vc\_VidCastTagList

WHERE vc\_VidCastTagList.vc\_TagID = @tagID

RETURN @returnValue END

GO

SELECT

vc\_Tag.TagText

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

FROM vc\_Tag

CREATE FUNCTION dbo.vc\_VidCastDuration(@userID int)

RETURNS int AS

BEGIN

DECLARE @returnValue int

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

WHERE vc\_VidCast.vc\_UserID = @userID

RETURN @returnValue

END

GO

SELECT

\*

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

FROM vc\_User

GO

CREATE VIEW vcTagReport AS

SELECT

vc\_Tag.TagText

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

FROM vc\_Tag

GO

SELECT \* FROM vcTagReport ORDER BY VidCasts DESC

ALTER VIEW vc\_MostProlificUsers AS

SELECT TOP 10

\*

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

,dbo.vc\_VidCastDuration(vc\_UserID) AS TptalMinutes

FROM vc\_User

ORDER BY VidCastCount DESC

GO

SELECT UserName, VidCastCount, TptalMinutes FROM vc\_MostProlificUsers

CREATE PROCEDURE vc\_AddTag2(@tagText varchar(20), @description varchar(100)=NULL) AS BEGIN

INSERT INTO vc\_Tag (TagText, TagDescription) VALUES(@tagText, @description)

RETURN @@identity

END

DECLARE @newTagID int

EXEC @newTagID = vc\_AddTag2 'SQL2', 'Finally, a SQL Tag'

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

CREATE PROCEDURE vc\_FinishVidCast(@vidCastID int) AS

BEGIN

UPDATE vc\_VidCast SET EndDateTime = GetDate(), vc\_StatusID = 3

WHERE vc\_VidCastID = @vidCastID

RETURN @vidCastID