

Beverlyn Tucker

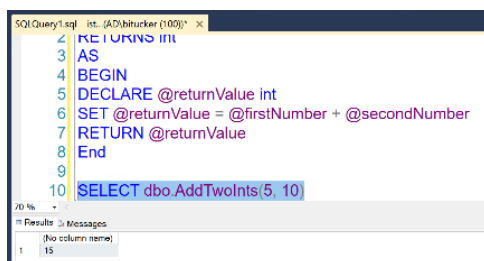
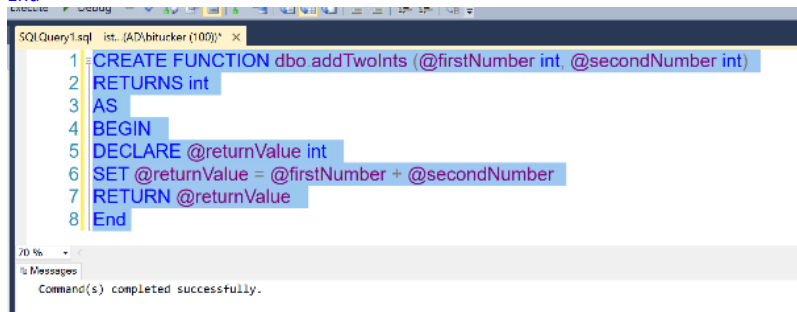
IST659_M406 Week8_Lab 08

Lab08 Database Programming

```
-- 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?
```

Results		Messages	
	(No column name)	(No column name)	
1	NULL	Yep, it is null	
	(No column name)	(No column name)	
1	Nope. It is not NULL	Nope. It is not NULL	

```
CREATE FUNCTION dbo.addTwoInts (@firstNumber int, @secondNumber int)
RETURNS int
AS
BEGIN
DECLARE @returnValue int
SET @returnValue = @firstNumber + @secondNumber
RETURN @returnValue
End
```



```
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

```

150 %

Results Messages

	vc_UserID	UserName	EmailAddress	UserDescription	WebSiteURL	UserRegisteredDate	VidCastCount
1	20	ecstatic	blandit.enim.consequat@loremutaliquam.co.uk	Bandwidth series A financing niche market.	NULL	2017-11-16 00:00:00.000	22
2	40	principle	ac.uma@miac.com	Business-to-business ecosystem ramen social media...	http://principle.vidcast659.site	2017-11-01 12:14:24.000	19
3	24	metacarpal	et.magna.Praesent@placeraaugueSed.org	Research & development startup long tail strategy g...	http://metacarpal.vidcast659.site	2017-05-30 11:31:12.000	18
4	43	canadian	Curabitur dictum.Phasellus@elefendnec.com	Agile development ownership business-to-consumer...	http://canadian.vidcast659.site	2017-06-27 05:16:48.000	18
5	26	przewalski	amet@Maurismolestie.org	NULL	http://przewalski.vidcast659.site	2017-02-11 08:52:48.000	17
6	36	silly	accumsan@gravidasagittisDuis.net	Stock founders early adopters low hanging fruit A/B...	http://silly.vidcast659.site	2017-05-25 14:38:24.000	17
7	7	wood	turpis.egestas.Fusce@massanonante.net	Technology investor marketing alpha.	http://wood.vidcast659.site	2017-06-21 15:36:00.000	16
8	11	doughnut	ipsum.primis@Cumsociis.com	Assets sales incubator user experience ecosystem ...	http://doughnut.vidcast659.site	2017-01-31 01:12:00.000	16
9	14	groggy	omare.In.faucibus@egestas.ca	Sales niche market user experience investor social ...	http://groggy.vidcast659.site	2017-04-20 09:50:24.000	16
10	47	these	parturient.montes@ipsum.ca	Entrepreneur virality freemium crowdsourcing long tail ...	http://these.vidcast659.site	2017-12-07 03:50:24.000	16

In your own words, in your answers document, describe what lines 1 through 5 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?

Answer: Line 1 through 5 does pulling top 10 userID that has the most VidCastCount and by using DESC given us the most to the fewest VidCastCount.

UserID 20 has a total of 22 VidCast how does the code know. Every time the user enters the new video it login the title of the video, and it counts every one of them.

```

GO
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')

```

150 %

Results Messages

(No column name)
1 4
(No column name)
1 NULL

In your own words, in your answers document, describe what lines 12 and 13 above actually do. Also, when line 13 executed, we received a NULL from SQL Server. How come?

Answer: What this means is when we are looking on the TagID lookup there are a total of 4 Music tag, and there was no Tunes tag.

```

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

```

	vc_UserID	UserName	EmailAddress	UserDescription	Web Site URL	UserRegisteredDate	VidCastCount
1	20	ecstatic	blandit.enim.consequat@loremutaliquam.co.uk	Bandwidth series A financing niche market.	NULL	2017-11-16 00:00:00.000	22
2	40	principle	ac.uma@miac.com	Business-to-business ecosystem ramen social media...	http://principle.vidcast659.site	2017-11-01 12:14:24.000	19
3	24	metacarpal	et.magna.Praesent@placeraaugueSed.org	Research & development startup long tail strategy g...	http://metacarpal.vidcast659.site	2017-05-30 11:31:12.000	18
4	43	canadian	Curabitur.dictum.Phasellus@eleifendnec.com	Agile development ownership business-to-consumer...	http://canadian.vidcast659.site	2017-06-27 05:16:48.000	18
5	26	przewalski	amet@Maurismolestie.org	NULL	http://przewalski.vidcast659.site	2017-02-11 08:52:48.000	17
6	36	silly	accumsan@gravidasagittisDuis.net	Stock founders early adopters low hanging fruit A/B...	http://silly.vidcast659.site	2017-05-25 14:38:24.000	17
7	7	wood	turpis.egestas.Fusce@massanonante.net	Technology investor marketing alpha.	http://wood.vidcast659.site	2017-06-21 15:36:00.000	16
8	11	doughnut	ipsum.primis@Cumsociis.com	Assets sales incubator user experience ecosystem ...	http://doughnut.vidcast659.site	2017-01-31 01:12:00.000	16
9	14	groggy	omare.In.faucibus@egestas.ca	Sales niche market user experience investor social ...	http://groggy.vidcast659.site	2017-04-20 09:50:24.000	16
10	47	these	parturient.montes@ipsum.ca	Entrepreneur virality freemium crowdsourcing long tail ...	http://these.vidcast659.site	2017-12-07 03:50:24.000	16

```

79 -- Create a view to retrieve the top 10 vc_Users and their
80 -- VidCast counts
81 CREATE VIEW vc_MostProlificUsers AS
82 SELECT TOP 10
83 *
84 , dbo.vc_VidCastCount(vc_UserID) as VidCastCount
85 FROM vc_User
86 ORDER BY VidCastCount DESC
87 GO

```

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

Answer: A view function is a mirror image of the real table in the database that contains the same information in the rows and column and adding Order by DESC it provides us information the most VidCastCount.

```

CREATE PROCEDURE vc_changeUserEmail(@username varchar(20), @newEmail varchar(50))
AS
BEGIN
    UPDATE vc_User SET EmailAddress = @newEmail
    WHERE UserName = @userName
END
GO

```

```

22 --Create a procedure to update a vc_User's email address
23 --The First parameter is the user name for the user to change
24 --The second is the new email address
25
26 CREATE PROCEDURE vc_changeUserEmail(@username varchar(20), @newEmail varchar(50))
27 AS
28 BEGIN
29     UPDATE vc_User SET EmailAddress = @newEmail
30     WHERE UserName = @userName
31 END
32 GO

```

Messages
Commands completed successfully.

```
EXEC vc_changeUserEmail 'tardy','kmstudent@syr.edu'

25 CREATE PROCEDURE vc_changeUserEmail(@username varchar(20), @newEmail varchar(50))
26 AS
27 BEGIN
28 UPDATE vc_User SET EmailAddress = @newEmail
29 WHERE UserName = @username
30 END
31 GO
32
33 EXEC vc_changeUserEmail 'tardy','kmstudent@syr.edu'
```

Messages

(1 row(s) affected)

```
SELECT * FROM vc_User
WHERE UserName = 'tardy'
```

```
35 SELECT * FROM vc_User
36 WHERE UserName = 'tardy'
```

150 %

	vc_UserID	UserName	EmailAddress	UserDescription	WebSiteURL	UserRegisteredDate
1	6	tardy	kmstudent@syr.edu	Startup leverage growth hacking bootstrapping sc...	http://tardy.videocast659.site	2017-03-12 15:36:00.000

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

Mine is Line 22 to 36 all the way down to Select Where statement

Answer: The green font is instruction on how to create a procedure and how to update.

The rest is the steps on how to code the procedure, and update User table under column name EmailAddress to the new email. Line 104, on mine, is line 35 & 36 this is a result after the updates it shows that the procedure creation and the update are successful.

```
INSERT INTO vc_Tag (TagText) VALUES ('cat Videos')
SELECT * FROM vc_Tag
WHERE vc_TagID = @@identity
```

```
38 INSERT INTO vc_Tag (TagText) VALUES ('cat Videos')
39 SELECT * FROM vc_Tag
40 WHERE vc_TagID = @@identity
41
```

150 %

	vc_TagID	TagText	TagDescription
1	15	cat Videos	NULL

```
CREATE 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 = @userName)
RETURN @@IDENTITY
END
GO
```

```

SQLQuery1.sql - ist...(AD/bitucker (106))
1 CREATE PROCEDURE vc_addUserLogin(@userName varchar(20), @loginFrom varchar(50))
2 AS
3 BEGIN
4 DECLARE @userID int
5 SELECT @userID = vc_UserID
6 FROM vc_User
7 WHERE UserName = @userName
8
9 INSERT INTO vc_UserLogin (vc_UserID,LoginLocation)
10 VALUES (@userID , @loginFrom)
11 RETURN @@IDENTITY
12 END
13 GO
70 %
Messages
Command(s) completed successfully.

```

```

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

```

100 %

Results Messages

	vc_UserID	UserName	UserLoginTimestamp	LoginLocation
1	6	tardy	2019-03-04 21:48:07.397	tardy

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

Answer: The UserLoginTimestamp is different from the attachment because it is based on the course when was created. Mine is base one time that I run the report/inserted the information.

On your answers doc, also identify one way we could simplify the code in the stored procedure above. (Hint: Look back at how we did a lookup with the vc_Tag table)

Answer: another way to get the same time is to hard code/declaring the return value. The UserLoginTimestamp. Means Insert the callout time.

Part 2 – Putting All Together

```

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

```

```

24 CREATE FUNCTION dbo.vc_UserIDLookup(@userName varchar(20))
25 RETURNS int AS
26 BEGIN
27     DECLARE @returnValue int
28     SELECT @returnValue = vc_userID FROM vc_User
29     WHERE userName = @userName
30
31
32     RETURN @returnValue
33 END
34 GO
35

```

Messages
Commands completed successfully.

SELECT 'Trying the vc_UserIDLookup function.', dbo.vc_UserIDLookup('tardy')

```

36 SELECT 'Trying the vc_UserIDLookup function.', dbo.vc_UserIDLookup('tardy')
37

```

150 %

Results		Messages	
(No column name)	(No column name)	(No column name)	(No column name)
1	Trying the vc_UserIDLookup function.	6	

```

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

```

	TagText	VidCasts
1	Art	256
2	Audio Recording	266
3	Baseball	242
4	Basketball	236
5	cat Videos	0
6	Collectibles	258
7	Consoles	260
8	Fashion	240
9	Football	259
10	Games	254
11	Motors	0
12	Music	237
13	Personal	263
14	Professional	264
15	Sports - General	235

```

CREATE FUNCTION vc_VidCastDuration(@UserID int)
RETURNS int AS
BEGIN
DECLARE @returnValue int
SELECT @returnValue = SUM(DATEDIFF(n, startDateTime, EndDateTime)) FROM vc_VidCast
Join vc_User on vc_User.vc_UserID = vc_VidCast.vc_UserID

```

WHERE vc_User.vc_UserID = @UserID

RETURN @returnValue

END

GO

Select

*,
dbo.vc_VidCastDuration(vc_UserID) AS TotalMinutes
FROM vc_User

vc_UserID	UserName	EmailAddress	UserDescription	WebSiteURL	UserRegisteredDate	TotalMinutes
1	ethanol	Donec.tempus@penatibusetmagnis.co.uk	Agile development non-disclosure agreement equity ...	http://ethanol.vidcast659.site	2017-12-30 22:19:12.000	1859
2	dispatcher	quam@aperttactisociosqu.ca	A/B testing handshake disruptive seed money info...	http://dispatcher.vidcast659.site	2017-12-08 03:36:00.000	1368
3	camel	mauris@massanon.edu	User experience founders branding entrepreneur iter...	http://camel.vidcast659.site	2017-08-14 03:21:36.000	1859
4	infatuated	mollis@Nam.org	Lean startup launch party angel investor branding b...	http://infatuated.vidcast659.site	2017-06-07 17:02:24.000	1426
5	hygienist	magna.Uk@necumasuscipit.ca	Business model canvas accelerator pivot network ef...	http://hygienist.vidcast659.site	2017-03-17 23:16:48.000	1139
6	tardy	kmstudent@syr.edu	Startup leverage growth hacking bootstrapping scru...	http://tardy.vidcast659.site	2017-03-12 15:36:00.000	2303
7	wood	turpis.egestas.Fusce@massanonante.net	Technology investor marketing alpha.	http://wood.vidcast659.site	2017-06-21 15:36:00.000	2292
8	mallard	vel.lectus.Cum@velteget.edu	Assets sales success bandwidth business model ca...	http://mallard.vidcast659.site	2017-09-15 19:55:12.000	1828
9	lifted	eu@elitsed.net	NULL	http://lifted.vidcast659.site	2017-04-15 20:24:00.000	1828
10	gum	ut@pharetraQuisqueac.com	Infographic incubator hypotheses client conversion ...	http://gum.vidcast659.site	2017-02-24 09:07:12.000	1238
11	doughnut	ipsum.primis@Cumsociis.com	Assets sales incubator user experience ecosystem a...	http://doughnut.vidcast659.site	2017-01-31 01:12:00.000	1830
12	bewildered	Donec.porttitor.tellus@odioAliquamvulputate.edu	Infrastructure research & development venture bum ...	http://bewildered.vidcast659.site	2017-12-29 09:07:12.000	1944
13	albite	nisi@vitaemauris.org	Learning curve partnership buzz value proposition re...	http://albite.vidcast659.site	2017-07-25 00:00:00.000	1971
14	groggy	omare.In.faucibus@egestas.ca	Sales niche market user experience investor social ...	http://groggy.vidcast659.site	2017-04-20 09:50:24.000	2464
15	bicycle	Quisque.porttitor.eros@mi.net	Success network effects focus monetization iPhone...	http://bicycle.vidcast659.site	2017-01-17 12:00:00.000	1152

SELECT

vc_Tag.TagText
,
dbo.vc_TagVidCastCount(vc_Tag.vc_TagID) AS VidCasts
FROM vc_Tag
ORDER BY VidCasts DESC

	TagText	VidCasts
1	Audio Recording	266
2	Professional	264
3	Personal	263
4	Consoles	260
5	Football	259
6	Collectibles	258
7	Art	256
8	Games	254
9	Baseball	242
10	Fashion	240
11	Music	237
12	Basketball	236
13	Sports - General	235
14	Motors	0
15	cat Videos	0

```

ALTER VIEW dbo.vc_MostProlificUsers AS
    SELECT TOP 10
        *
        , dbo.vc_VidCastCount(vc_UserID) AS VidCastCount
        , dbo.vc_VidCastDuration(vc_UserID) AS TotalMinutes
    FROM vc_User

ORDER BY VidCastCount DESC

GO

```

```

SELECT UserName, VidCastCount, TotalMinutes FROM vc_MostProlificUsers

```

Results Messages			
	UserName	VidCastCount	TotalMinutes
1	ecstatic	22	2682
2	principle	19	3413
3	canadian	18	1928
4	metacarpal	18	3053
5	przewalski	17	2664
6	silly	17	2851
7	archives	16	2374
8	doughnut	16	1830
9	groggy	16	2464
10	sines	16	2316

```

Create PROCEDURE vc_addTagText(@TagText varchar(20), @description varchar(100)=NULL)
AS
BEGIN
    DECLARE @TagID int
    SELECT @TagID = vc_TagID
    FROM vc_Tag
    WHERE TagText = @TagText

    INSERT INTO vc_Tag (TagText, Tagdescription)
    VALUES (@TagText, @description)
    RETURN @@IDENTITY
END
GO

```



```
3 Create PROCEDURE vc_addTagText(@TagText varchar(20), @description varchar(100)=NULL)
4 AS
5 BEGIN
6     DECLARE @TagID int
7     SELECT @TagID = vc_TagID
8     FROM vc_Tag
9     WHERE TagText = @TagText
10
11 INSERT INTO vc_Tag (TagText, Tagdescription)
12 VALUES (@TagText, @description)
13 RETURN @@IDENTITY
14 END
15 GO
```

50 %

Messages

Commands completed successfully.

```
Create PROCEDURE vc_addTag(@TagText varchar(20), @description varchar(100)=NULL)
AS
BEGIN
DECLARE @TagID int
SELECT @TagID = vc_TagID
FROM vc_Tag
WHERE TagText = @TagText
```

```
INSERT INTO vc_Tag (TagText, Tagdescription)
VALUES (@TagText, @description)
RETURN @@IDENTITY
END
GO
```

```
DECLARE @newTagID int
EXEC @newTagID = vc_addTag 'SQL', 'Finally, a SQL Tag'
SELECT * FROM vc_Tag
WHERE vc_TagID = @newTagID
```

Results		Messages	
	vc_TagID	TagText	TagDescription
1	17	SQL	Finally, a SQL Tag

```
Create PROCEDURE vc_FinishVidCast(@vc_VidCastID int)
AS
BEGIN
DECLARE @VidCastID int
SELECT @vc_VidCastID = vc_VidCastID
FROM vc_VidCast
WHERE @vc_VidCastID = vc_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 = @@identity
SELECT * FROM vc_VidCast WHERE vc_VidCastID = @newVC
EXEC vc_FinishVidCast @newVC
SELECT * FROM vc_VidCast WHERE vc_VidCastID = @newVC

```

	vc_VidCastID	VidCastTitle	StartDateTime	EndDateTime	ScheduleDurationMinutes	RecordingURL	vc_UserID	vc_StatusID
1	860	Finally done with sprocs	2019-03-08 15:22:45.453	NULL	45	NULL	6	2

	vc_VidCastID	VidCastTitle	StartDateTime	EndDateTime	ScheduleDurationMinutes	RecordingURL	vc_UserID	vc_StatusID
1	860	Finally done with sprocs	2019-03-08 15:22:45.453	NULL	45	NULL	6	2

SQL Code

```

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')

```

```

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 vc_VidCastDuration(@UserID int)
RETURNS int AS

```

```

BEGIN
DECLARE @returnValue int
SELECT @returnValue = SUM(DATEDIFF(n, startDateTime, EndDateTime)) FROM vc_VidCast
Join vc_User on vc_User.vc_UserID = vc_VidCast.vc_UserID
WHERE vc_User.vc_UserID = @UserID

RETURN @returnValue
END
GO

```

```

Select
    *
    , dbo.vc_VidCastDuration(vc_UserID) AS TotalMinutes
FROM vc_User

```

```

SELECT
    vc_Tag.TagText
    , dbo.vc_TagVidCastCount(vc_Tag.vc_TagID) AS VidCasts
FROM vc_Tag
ORDER BY VidCasts DESC

```

```

ALTER VIEW dbo.vc_MostProlificUsers AS
    SELECT TOP 10
        *
        , dbo.vc_VidCastCount(vc_UserID) AS VidCastCount
        , dbo.vc_VidCastDuration(vc_UserID) AS TotalMinutes
FROM vc_User

```

```

ORDER BY VidCastCount DESC

```

```

GO
SELECT UserName, VidCastCount, TotalMinutes FROM vc_MostProlificUsers

```

```

Create PROCEDURE vc_addTagText(@TagText varchar(20), @description varchar(100)=NULL)
AS
BEGIN
DECLARE @TagID int
SELECT @TagID = vc_TagID
FROM vc_Tag
WHERE TagText = @TagText

INSERT INTO vc_Tag (TagText, Tagdescription)
VALUES (@TagText, @description)
RETURN @@IDENTITY
END
GO

```

```

Create PROCEDURE vc_addTag(@TagText varchar(20), @description varchar(100)=NULL)
AS
BEGIN
DECLARE @TagID int
SELECT @TagID = vc_TagID
FROM vc_Tag
WHERE TagText = @TagText

INSERT INTO vc_Tag (TagText, Tagdescription)
VALUES (@TagText, @description)
RETURN @@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(@vc_VidCastID int)
AS
BEGIN
DECLARE @VidCastID int
SELECT @vc_VidCastID = vc_VidCastID
FROM vc_VidCast
WHERE @vc_VidCastID = vc_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 = @@identity
SELECT * FROM vc_VidCast WHERE vc_VidCastID = @newVC
EXEC vc_FinishVidCast @newVC
SELECT * FROM vc_VidCast WHERE vc_VidCastID = @newVC
```