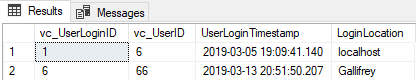
**Part 1—Securing Data Objects**

**User Login:**

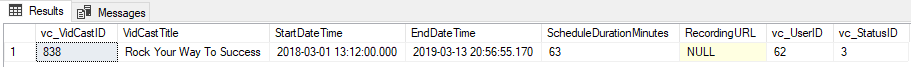
--Creating a guestuser database user

CREATE USER guestuser FOR LOGIN guestuser

**vc\_UserLogin:**



**vc\_VidCast Modified by guestuser:**



**Regular Tab:**

--Creating a guestuser database user

CREATE USER guestuser FOR LOGIN guestuser

--Grant read permission on the user table

GRANT SELECT ON vc\_User to guestuser

--Revoke the select permission!

REVOKE SELECT ON vc\_User to guestuser

--Give them the view instead

GRANT SELECT ON vc\_MostProlificUsers to guestuser

--Allow guestuser to run some stored procedures

GRANT EXECUTE ON vc\_AddUserLogin TO guestuser

GRANT EXECUTE ON vc\_FinishVidCast TO guestuser

--Retrieve all rows from the vc\_UserLogin table

SELECT \* FROM vc\_UserLogin

SELECT \* FROM vc\_VidCast WHERE vc\_VidCastID = '838'

**guestuser Tab:**

--Guestuser's tab

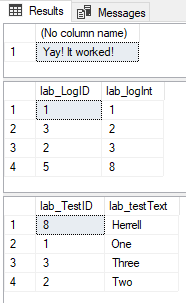
SELECT \* FROM vc\_User

SELECT \* FROM vc\_MostProlificUsers

EXEC dbo.vc\_AddUserLogin 'TheDoctor', 'Gallifrey'

EXEC dbo.vc\_FinishVidCast '838'

**Part 2—Data Integrity Through Transactions**



Yes, this transaction to add my last name worked. The first execution did not work because the value ‘One’ already existed in the table and we told our database in the previous code not to allow duplicate values. I expected the failure message I received on the first execution.

**SQL:**

--Creating a new table

CREATE TABLE lab\_Test (

lab\_TestID int identity primary key

, lab\_testText varchar(20) unique not null

)

/\*

This will be a table to keep log of

created lab\_Testrecords.

We don't want to add a row to this if

the inserty into lab\_Test fails

\*/

CREATE TABLE lab\_Log(

lab\_LogID int identity primary key

, lab\_logInt int unique not null

)

INSERT INTO lab\_Test (lab\_testText) VALUES ('One'), ('Two'), ('Three')

INSERT INTO lab\_Log (lab\_logInt) SELECT lab\_TestID FROM lab\_Test

--Step 1: Begin the transaction

BEGIN TRANSACTION

--Step 2: Assess the state of things.

DECLARE @rc int

SET @rc = @@ROWCOUNT --Initially 0

--Step 3: Make the change

--On success, @@ROWCOUNT is incremented by 1

--On failure, @@ROWCOUNT does not change

INSERT INTO lab\_Test (lab\_testText) VALUES ('Herrell')

--Step 4: Check the new state of things

IF(@rc = @@ROWCOUNT) -- If @@ROWCOUNT was not changed, fail

BEGIN

--Step 5, if failed

SELECT 'Bail out! It Failed!'

ROLLBACK

END

ELSE --Success! Continue.

BEGIN

--Step 5, if succeeded

SELECT 'Yay! It worked!'

INSERT INTO lab\_log (lab\_logInt) VALUES (@@identity)

COMMIT

END

SELECT \* FROM lab\_Log

SELECT \* FROM lab\_Test