Nick Videtti

IST-659 Week 9 Lab Assignment

**PART 1:**

***--Creating a guestuser database user***

***CREATE USER guestuser FOR LOGIN guestuser***

|  |
| --- |

|  |
| --- |

**My Tab:**

/\*

Author: Nick Videtti

Course : IST659 M407

Term : Spring (April - June) 2022

\*/

----------LAB 9----------

-----PART 1-----

--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

--Find VidCastID for 'Rock Your Way To Success' SELECT

vc\_VidCastID

FROM

vc\_VidCast

WHERE

VidCastTitle = 'Rock Your Way To Success'

SELECT

\*

FROM

vc\_UserLogin

SELECT

\*

FROM

vc\_VidCast

WHERE

VidCastTitle = 'Rock Your Way To Success'

**guestuser Tab:**

USE IST659\_M407\_nvidetti

--Guestuser's tab

SELECT \* FROM vc\_User

SELECT

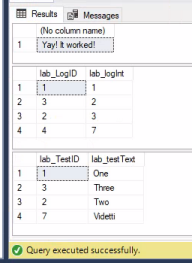
\*

FROM

vc\_MostPorlificUsers

EXEC vc\_AddUserLogin 'TheDoctor', 'Gallifrey' EXEC vc\_FinishVidCast 838

**PART 2:**

****

The reason that the execution failed at first is because we tried to enter a value of ‘One’ for lab\_testTest, but that violated the UNIQUE constraint on the lab\_testText column. Since my last name did not already exist as a value for lab\_testText, the second execution did not fail.

**Part 2 SQL Code:**

REVOKE SELECT ON vc\_MostProlificUsers TO guestUser

REVOKE EXECUTE ON vc\_AddUserLogin TO guestuser

REVOKE EXECUTE ON vc\_FinishVidCast TO guestuser

-----PART 2-----

--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 a log of created lab\_Test records. We don't want to add a row to this if the insert 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 ('Videtti')

--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