|  |  |  |
| --- | --- | --- |
| **Name** | : | Sathish Kumar Rajendiran |
| **Current Date** | : | 03/21/2020 |
| **Lab Number** | : | 09 |

**Part 1 – Introducing Functions, Securing Data Objects**

* ***Creating SQL Login user - guestuser***

/\* Creating a Database User

1. Creating a guestuser database user \*/

-- Check SQL Server Login

IF SUSER\_ID('guestuser') IS NULL

CREATE LOGIN guestuser WITH PASSWORD = 'SU2orange!';

-- Check database user

IF USER\_ID('guestuser') IS NULL

CREATE USER guestuser FOR LOGIN guestuser;

/\*

* ***Login as guestuser***

**A screenshot of a cell phone

Description automatically generated**

2. Allow user guestuser EXECUTE permission to the STORED PROCEDURES \*/

GRANT EXECUTE ON dbo.vc\_AddUserLogin to guestuser

GRANT EXECUTE ON dbo.vc\_FinishVidCast to guestuser

GRANT SELECT ON dbo.vc\_VidCast to guestuser

/\*

EXECUTE the STORED PROCEDURES on guestuser tab

3. Execute Stored Procedure to Add records to vc\_UserLogin table \*/

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

/\*

4. Execute Stored Procedure to Update vc\_VidCast table \*/

IF EXISTS (select distinct VidCastTitle from dbo.vc\_VidCast where VidCastTitle ='Rock Your Way To Success')

BEGIN

DECLARE @vc\_VidCastID int

Select @vc\_VidCastID = vc\_VidCastID from dbo.vc\_VidCast where VidCastTitle ='Rock Your Way To Success'

---Print @vc\_VidCastID

EXEC vc\_FinishVidCast @vc\_VidCastID

END

/\* Retrieve records from srajendi tab. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

----5. Select all records from dbo.vc\_UserLogin table

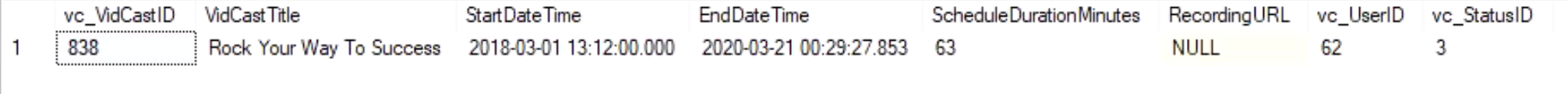
SELECT \* from dbo.vc\_UserLogin

**A screenshot of a cell phone

Description automatically generated**

--- 6. Select all modified records from vc\_VidCast by guestuser \*/

SELECT \* from dbo.vc\_VidCast where VidCastTitle ='Rock Your Way To Success'



**Part 2 – Data Integrity though Transactions**

* ***Successful entry into dbo.lab\_Test table***

A screenshot of a cell phone

Description automatically generated

* ***While data entry violated unique constraints on dbo.lab\_Test table***

A screenshot of a cell phone

Description automatically generated

**Explanation:**

Whole statement is executed in a controlled transaction mode. In addition, I have also added TRY & CATCH blocks to capture the exceptions.

* + Table dbo.lab\_Log has Unique constraint on lab\_TestText field.
  + Table dbo.lab\_Test has Unique constraint on lab\_TestInt field.
  + insert statement failed because of the Unique constraint violation.

**SQL Scripts from AD\srajendi tab:**

USE [IST659\_M406\_srajendi]

GO

/\*

Title : Lab 09 - Data Security - SQL Scripts

Author : Sathish Kumar Rajendiran

Course : IST659 M406

Term : March 2020

User : AD\srajendi / Windows Authentication AD User

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 1 - Securing Data Objects \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Creating a Database User

1. Creating a guestuser database user \*/

-- Check SQL Server Login

IF SUSER\_ID('guestuser') IS NULL

CREATE LOGIN guestuser WITH PASSWORD = 'SU2orange!';

-- Check database user

IF USER\_ID('guestuser') IS NULL

CREATE USER guestuser FOR LOGIN guestuser;

/\* Managing user permissions

2. Grant user guestuser READ permission to the dbo.vc\_User TABLE \*/

GRANT SELECT ON dbo.vc\_User to guestuser

/\*

3. REVOKE guestuser's access from dbo.vc\_User TABLE \*/

REVOKE SELECT ON dbo.vc\_User to guestuser

/\*

4. Grant user guestuser READ permission to the dbo.vc\_MostProlificUsers TABLE \*/

GRANT SELECT ON dbo.vc\_MostProlificUsers to guestuser

/\*

5. Allow user guestuser EXECUTE permission to the STORED PROCEDURES \*/

GRANT EXECUTE ON dbo.vc\_AddUserLogin to guestuser

GRANT EXECUTE ON dbo.vc\_FinishVidCast to guestuser

GRANT SELECT ON dbo.vc\_VidCast to guestuser -- to be able to set query valdiation; Optional statement included only to make the execute statement repeateable.

/\* Retrive records

6. Select all records from dbo.vc\_UserLogin table \*/

SELECT \* from dbo.vc\_UserLogin

/\*

7. Select all modified records from vc\_VidCast by guestuser \*/

Select \* from dbo.vc\_VidCast where VidCastTitle ='Rock Your Way To Success'

/\* REVOKE Permissions for guestuser on all objects

8. REVOKE Permission on Strored Procedure, View and Tables \*/

REVOKE SELECT ON vc\_MostProlificUsers to guestuser

REVOKE SELECT ON vc\_UserLogin to guestuser

REVOKE EXECUTE ON vc\_AddUserLogin TO guestuser

REVOKE EXECUTE ON vc\_FinishVidCast TO guestuser

/\*\*\*\*\*\*\* Part 2: Data Intergrity through Transactions\*/

IF EXISTS (SELECT \* FROM INFORMATION\_SCHEMA.TABLES WHERE TABLE\_NAME='lab\_Test')

BEGIN

DROP TABLE lab\_Test

END

GO

IF EXISTS (SELECT \* FROM INFORMATION\_SCHEMA.TABLES WHERE TABLE\_NAME='lab\_Log')

BEGIN

DROP TABLE lab\_Log

END

GO

-----Creating the lab\_Test table

CREATE TABLE lab\_Test (

--Columns for the lab\_Test table

lab\_TestID int identity PRIMARY KEY,

lab\_TestText varchar(20) not null CONSTRAINT U1\_lab\_Test UNIQUE (lab\_TestText)

)

GO

-----Creating the lab\_Log table

---- This table will keep a log of created lab\_Test records.

---- We dont want to add new records if thier insert fails into lab\_Test table.

CREATE TABLE lab\_Log (

--Columns for the lab\_Log table

lab\_LogID int identity PRIMARY KEY,

lab\_TestInt int not null CONSTRAINT U1\_lab\_Log UNIQUE (lab\_TestInt)

)

GO

----Adding Data to lab\_Test & lab\_Log tables

INSERT INTO lab\_Test(lab\_TestText)

VALUES

('One'),('Two'),('Three')

GO

INSERT INTO lab\_Log(lab\_TestInt)

SELECT lab\_TestID

From dbo.lab\_Test

GO

BEGIN TRY

BEGIN TRANSACTION

Declare @rc int

SET @rc = @@ROWCOUNT

IF NOT EXISTS ( Select count(\*) from dbo.lab\_Test where lab\_TestText ='srajendi' )

BEGIN

SET @rc = @@ROWCOUNT +1

END

INSERT into dbo.lab\_Test(lab\_TestText) VALUES('srajendi')

----Step 4: Check the new state of things

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

BEGIN

SELECT 'Bail out! It Failed!'

ROLLBACK

END

ELSE ----Success! Continue.

BEGIN

----Step 5, if Succeeded

SELECT 'Yay! It Worked!'

INSERT INTO lab\_Log(lab\_TestInt) VALUES (@@identity)

COMMIT

END

END TRY

BEGIN CATCH

ROLLBACK TRANSACTION

SELECT 'Bail out! It Failed!'

END CATCH

SELECT \* FROM lab\_Log go;

SELECT \* FROM lab\_Test go;

/\*--------------------------------------------------------------

**SQL Scripts from guestuser tab**

------------------------------------------------------------------------\*/

USE [IST659\_M406\_srajendi]

GO

/\*

Title : Lab 09 - Data Security - SQL Scripts

Author : Sathish Kumar Rajendiran

Course : IST659 M406

Term : March 2020

User : guestuser / SQL User

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 1 - Securing Data Objects \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

--- First SQL Querying for guestuser

Select \* from dbo.vc\_User ---- fails as object level access not been granted yet.

--- After the GRANT statement on the vc\_User's table for guestuser

Select \* from dbo.vc\_User ---- returns data from vc\_User's table.

--- After the REVOKE statement on the vc\_User's table for guestuser

Select \* from dbo.vc\_User --- fails as object level access has been revoked.

/\*

Msg 229, Level 14, State 5, Line 21

The SELECT permission was denied on the object 'vc\_User', database 'IST659\_M406\_srajendi', schema 'dbo'.

\*/

--- After the GRANT statement on the vc\_MostProlificUsers's View for guestuser

Select \* from dbo.vc\_MostProlificUsers ---- returns data from vc\_MostProlificUsers's view.

/\*

EXECUTE the STORED PROCEDURES

Execute Stored Procedure to Add records to vc\_UserLogin table \*/

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

/\*

Execute Stored Procedure to Update vc\_VidCast table \*/

IF EXISTS( select distinct VidCastTitle from dbo.vc\_VidCast where VidCastTitle ='Rock Your Way To Success')

BEGIN

DECLARE @vc\_VidCastID int

Select @vc\_VidCastID = vc\_VidCastID from dbo.vc\_VidCast where VidCastTitle ='Rock Your Way To Success'

---Print @vc\_VidCastID

EXEC vc\_FinishVidCast @vc\_VidCastID

END