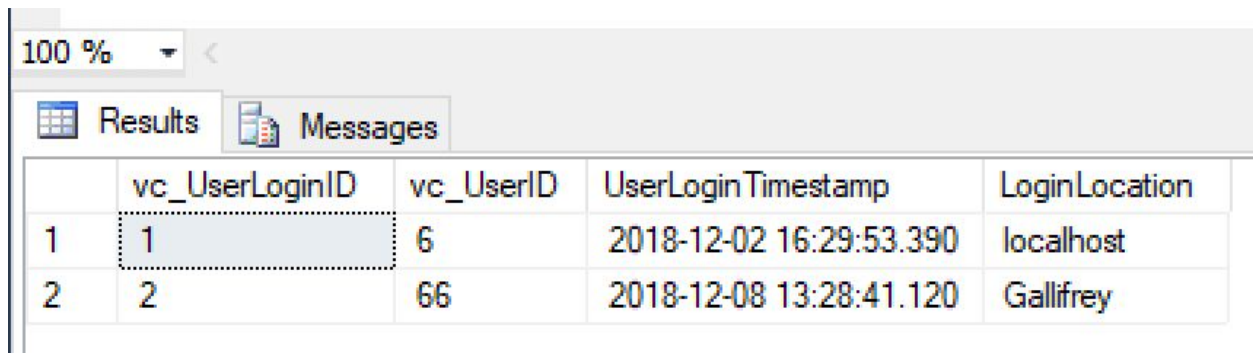


Kendra Osburn
December 8, 2018
Lab 09

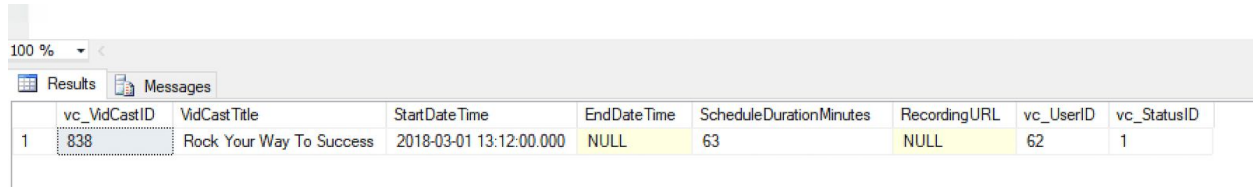
CREATING A DATABASE USER

-- Creating a guest user for database user
CREATE USER guestuser FOR LOGIN guestuser



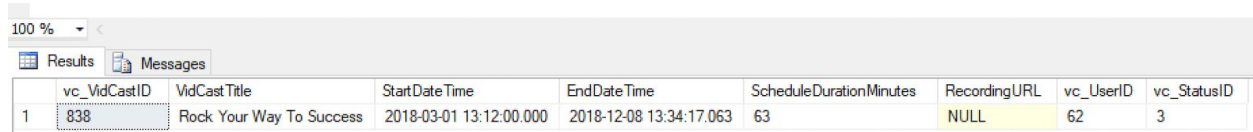
	vc_UserLoginID	vc_UserID	UserLoginTimestamp	LoginLocation
1	1	6	2018-12-02 16:29:53.390	localhost
2	2	66	2018-12-08 13:28:41.120	Gallifrey

BEFORE GUESTUSER "FINISHED" VIDCAST



	vc_VidCastID	VidCastTitle	StartDateTime	EndDateTime	ScheduleDurationMinutes	RecordingURL	vc_UserID	vc_StatusID
1	838	Rock Your Way To Success	2018-03-01 13:12:00.000	NULL	63	NULL	62	1

AFTER GUESTUSER "FINISHED" VIDCAST



	vc_VidCastID	VidCastTitle	StartDateTime	EndDateTime	ScheduleDurationMinutes	RecordingURL	vc_UserID	vc_StatusID
1	838	Rock Your Way To Success	2018-03-01 13:12:00.000	2018-12-08 13:34:17.063	63	NULL	62	3

GUESTUSER CODE

```
SELECT * FROM vc_MostProlificUsers  
EXEC vc_AddUserLogin 'TheDoctor', 'Gallifrey'  
EXEC vc_FinishVidcast 838
```

MY CODE

```
-- Creating a guest user for database user  
CREATE USER guestuser FOR LOGIN guestuser
```

```
SELECT * FROM vc_User
```

```
-- revoke select permissions!
```

```
REVOKE SELECT ON vc_User to guestuser
```

```
--Grant read permission on the user table
GRANT SELECT ON vc_MostProlificUsers to guestuser
```

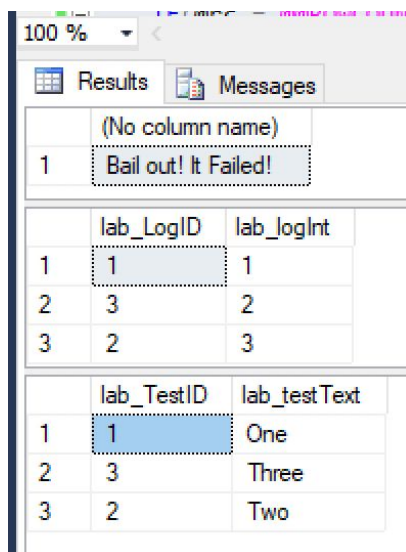
```
-- allow guestuser to run some stored procedures
GRANT EXECUTE ON vc_AddUserLogin TO guestuser
GRANT EXECUTE ON vc_FinishVidCast TO guestuser
```

```
SELECT * FROM vc_UserLogin
```

```
-- finding info about this clearly revolutionary vidcast
SELECT * FROM vc_VidCast WHERE VidCastTitle LIKE 'Rock Your Way To Success'
```

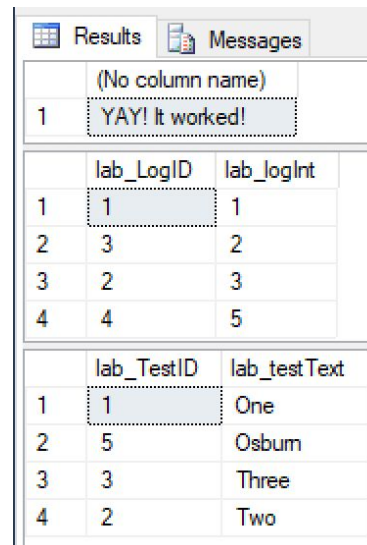
```
-- revoking for safety
REVOKE EXECUTE ON vc_AddUserLogin to guestuser
REVOKE EXECUTE ON vc_FinishVidCast to guestuser
```

FIRST



	lab_LogID	lab_logInt
1	1	1
2	3	2
3	2	3

SECOND



	lab_LogID	lab_logInt
1	1	1
2	3	2
3	2	3
4	4	5

IN MY OWN WORDS, WHY DID ONE FAIL AND TWO SUCCEED?

In the first attempt, we were trying to insert a value that already existed into the database, so it threw an error and didn't execute the rest of the statement -- it BAILED OUT!!

In the second attempt, we inserted a new value so when it got to the "did this transaction succeed" line, the code said "YES!!"

FINAL SQL

```
-- Creating a guest user for database user
CREATE USER guestuser FOR LOGIN guestuser
```

```
SELECT * FROM vc_User
```

```
-- revoke select permissions!
```

```

REVOKE SELECT ON vc_User to guestuser

--Grant read permission on the user table
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

SELECT * FROM vc_UserLogin

-- finding info about this clearly revolutionary vidcast
SELECT * FROM vc_VidCast WHERE VidCastTitle LIKE 'Rock Your Way To Success'

REVOKE EXECUTE ON vc_AddUserLogin to guestuser
REVOKE EXECUTE ON vc_FinishVidCast to guestuser

-- Creating a new table
CREATE TABLE lab_Test (
    lab_TestID int identity primary key,
    lab_testText varchar(20) unique not null
)

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 TRANSACTION
BEGIN TRANSACTION
    -- STEP 2 ASSESS
    DECLARE @rc int
    SET @rc = @@ROWCOUNT -- Initially 0

    -- STEP 3 MAKE CHANGE
    INSERT INTO lab_Test (lab_testText) VALUES ('Osburn')

    -- STEP 4 CHECK 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
```