Creating a database user:
CREATE USER guestuser FOR LOGIN guestuser

--Guestuser's tab
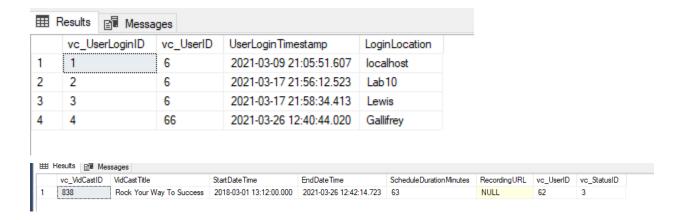SELECT * FROM vc_User
--ProlificUsers
SELECT * FROM vc_MostProlificUsers
EXEC vc_AddUserLogin 'TheDoctor', 'Gallifrey'
EXEC vc_FinishVidcast 838

--MyTab
--Grant read permission on the user table
GRANT SELECT ON vc_User to guestuser
--Revoke on 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

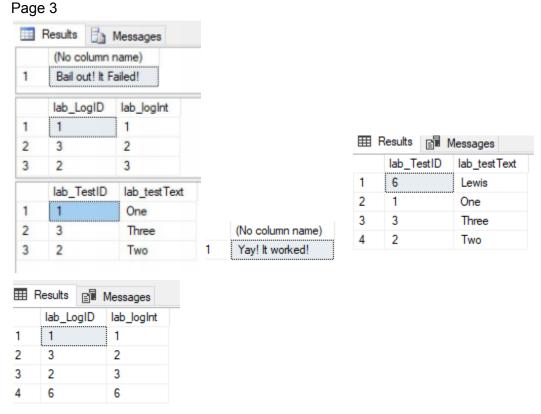Select * from vc_VidCast where VidCastTitle like 'Rock Your Way To Success'

Select * from vc_UserLogin

	vc_UserLoginID	vc_UserID	UserLoginTimestamp	LoginLocation
1	1	6	2021-03-09 21:05:51.607	localhost
2	2	6	2021-03-17 21:56:12.523	Lab10
3	3	6	2021-03-17 21:58:34.413	Lewis
4	4	66	2021-03-26 12:40:44.020	Gallifrey

	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	2021-03-26 12:42:14.723	63	NULL	62	3

Part 2
--Creating a new table
CREATE TABLE lab_Test (
        lab_TestID int identity primary key,
        lab_testText varchar(20) unique not null
)

```sql
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 ('One')

        --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
```

Results | **Messages**

	(No column name)
1	Bail out! It Failed!

	lab_LogID	lab_logInt
1	1	1
2	3	2
3	2	3

	lab_TestID	lab_testText
1	1	One
2	3	Three
3	2	Two

	(No column name)
1	Yay! It worked!

Results | **Messages**

	lab_TestID	lab_testText
1	6	Lewis
2	1	One
3	3	Three
4	2	Two

Results | **Messages**

	lab_LogID	lab_logInt
1	1	1
2	3	2
3	2	3
4	6	6

We were trying to insert a value that already existed in the database in the first attempt, so it threw an error and didn't finish the sentence — it BAILED OUT!!
We inserted a new value in the second attempt, and the code said "YES!!" when it got to the "did this transaction succeed" section.