

Unit 11 Problem Set Submission Form

Overview

Your Name	Bhavya shah
Your SU Email	bhshah@syr.edu

Instructions

Put your name and SU email at the top. Answer these questions all from the lab. When asked to include screenshots, please follow the screen shot guidelines from the first lab.

Remember as you complete the problem sets it is not only about getting it right / correct. We will discuss the answers in class so it's important to articulate anything you would like to contribute to the discussion in your answer:

- If you feel the question is vague, include any assumptions you've made.
- If you feel the answer requires interpretation or justification provide it.
- If you do not know the answer to the question, articulate what you tried and how you are stuck.

This how you receive credit for answering questions which might not be correct.

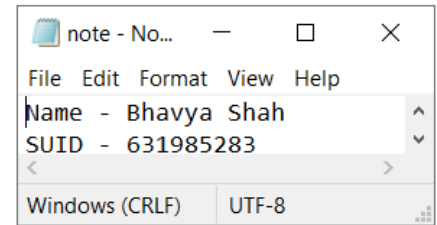
Questions

Answer these questions using the problem set submission template. You will need to consult the logical model in the overview section for details. For any screenshots provided, please follow the guidelines for submitting a screenshot.

Write the following as SQL programs. For each, include the SQL as a screenshot with the output of the SQL Code.

1. Provide a screenshot of your code execution from the walkthrough were you modified **p_upsert_major** in the **TinyU** database to be transaction-safe.

```
1 DROP PROCEDURE if EXISTS [dbo].[p_upsert_major]
2 Go
3 create PROCEDURE [dbo].[p_upsert_major]
4 (
5     @major_id as int,
6     @major_code as VARCHAR(20),
7     @major_name as VARCHAR(100)
8 ) as
9 BEGIN
10     BEGIN TRY
11         begin tran X
12         IF not exists (SELECT major_code from majors where major_id= @major_id) BEGIN
13             INSERT into majors (major_id, major_code, major_name)
14             VALUES (@major_id, @major_code, @major_name)
15             if @@ROWCOUNT <> 1 THROW 5001, 'p_upsert_major:Insert Error', 1
16         END
17         else BEGIN
18             UPDATE majors
19             set major_name=@major_name
20             WHERE major_id =@major_id
21             if @@ROWCOUNT <>1 THROW 5002 , 'p_upsert_major:Update Error', 2
22         end
23         COMMIT TRAN x
24     end try
25     begin catch
26         ROLLBACK tran x
27     ;
28     THROW
29     end
30 CATCH
31 end
32 go
```



Messages

```
12:57:35 PM Started executing query at Line 1
Commands completed successfully.
12:57:35 PM Started executing query at Line 3
Commands completed successfully.
Total execution time: 00:00:00.091
```

2. Provide a screenshot of examples of executing the **p_upsert_major** procedure to demonstrate it is transaction safe.

```

1 select * from majors
2 go
3 exec [dbo].[p_upsert_major] @major_id=5, @major_code='BSDD', @major_name='Basket Weaving'
4 go
5 select * from majors

```

major_id	major_code	major_name
1	IMT	Information Management and T...
2	ADS	Applied Data Science
3	ACC	Accounting
4	CSC	Computer Sciences
5	BSK	Basket Weaving

major_id	major_code	major_name
1	IMT	Information Management and T...
2	ADS	Applied Data Science
3	ACC	Accounting
4	CSC	Computer Sciences
5	BSK	Basket Weaving

note - No...

Name - Bhavya Shah
SUID - 631985283

Windows (CRLF) UTF-8

3. Re-write the **p_place_bid** stored procedure from the **vBay** database so that it is transaction safe. Provide a screenshot of the code and its execution.

```

1 GO
2 DROP procedure if EXISTS dbo.p_place_bid
3 GO
4 create procedure [dbo].[p_place_bid]
5 (
6     @bid_item_id int,
7     @bid_user_id int,
8     @bid_amount money
9 )
10 as begin
11     declare @max_bid_amount money
12     DECLARE @item_seller_user_id int
13     DECLARE @bid_status VARCHAR(20)
14     begin TRANSACTION
15     begin try
16         set @bid_status='ok'
17         set @max_bid_amount=(select max (bid_amount) from vb bids where bid_item_id=@bid_item_id and bid_status='ok')
18         set @item_seller_user_id=(select item_seller_user_id from vb items where item_id=@bid_item_id)
19         if (@max_bid_amount is null)
20             set @max_bid_amount = (select item_reserve from vb items where item_id= @bid_item_id)
21         if (@item_seller_user_id=@bid_user_id)
22             set @bid_status='item_seller'
23         if (@bid_amount<= @max_bid_amount)
24             set @bid_status = 'low_bid'
25         insert into vb_bids (bid_user_id, bid_item_id, bid_amount, bid_status)
26         VALUES (@bid_user_id, @bid_item_id, @bid_amount, @bid_status)
27         PRINT 'transaction count is '+ cast(@@trancount as varchar(20))
28         COMMIT
29         PRINT 'comitting'
30         PRINT 'transaction count is '+ cast (@@trancount as varchar (20))
31     end try
32     begin catch
33         ROLLBACK;
34         THROW
35     end CATCH
36 END

```

note - No...

Name - Bhavya Shah
SUID - 631985283

Windows (CRLF) UTF-8

Messages

1:23:17 PM Started executing query at Line 2
Commands completed successfully.

1:23:17 PM Started executing query at Line 4
Commands completed successfully.
Total execution time: 00:00:00.130

4. Execute your stored procedure in step 3 to demonstrate the procedure works. Make user 2, Bid \$105 on item 36 and show the bid was placed with a SELECT.

Run Cancel Disconnect Change Connection vbay Estimated Plan Enable Actual Plan

```

1 execute p_place_bid @bid_amount= '$105', @bid_item_id= 36, @bid_user_id=2
2 GO
3 select * from vb_bids where bid_item_id=36

```

note - No...
 File Edit Format View Help
 Name - Bhavya Shah
 SUID - 631985283
 Windows (CRLF) UTF-8

Results Messages

	bid_id	bid_user_id	bid_item_id	bid_datetime	bid_amount	bid_status
1	70	1	36	2022-09-10 03:36:15.487	80.00	ok
2	71	2	36	2022-09-10 03:36:15.487	85.00	ok
3	72	1	36	2022-09-10 03:36:15.487	90.00	ok
4	73	2	36	2022-09-10 03:36:15.487	95.00	ok
5	74	1	36	2022-09-10 03:36:15.487	95.00	low_bid
6	75	1	36	2022-09-10 03:36:15.487	100.00	ok
7	1002	2	36	2022-11-14 18:27:13.737	105.00	ok

5. Re-write the **p_rate_user** stored procedure from the **VBay** database so that it is transaction safe. Provide a screenshot of the code and its execution.

```

1 drop PROCEDURE if exists p_rate_user
2 go
3 create PROCEDURE [dbo].[p_rate_user]
4 (
5     @rating_by_user_id int,
6     @rating_for_user_id int,
7     @rating_astype varchar(20),
8     @rating_value int,
9     @rating_comment text
10 ) as
11 BEGIN
12 begin TRANSACTION
13 begin TRY
14 insert into vb user_ratings(rating by user id, rating for user id, rating astype, rating value, rating comment)
15 VALUES(@rating_by_user_id, @rating_for_user_id, @rating_astype, @rating_value, @rating_comment)
16 commit
17 return @@identity
18 end try
19 begin CATCH
20 select ERROR_NUMBER() as error, ERROR_MESSAGE() as msg
21 print 'transaction count is ' + cast(@@trancount as varchar(20))
22 if not exists (select * from vb user_ratings where rating_value>5)
23 throw 50001, 'Raing should be in between 1 to 5 ',1
24 ROLLBACK
25 Print 'rollback'
26 print 'transaction count is ' + cast(@@trancount as varchar(20))
27 end catch
28 END

```

note - No...
 File Edit Format View Help
 Name - Bhavya Shah
 SUID - 631985283
 Windows (CRLF) UTF-8

Messages

```

1:00:07 PM Started executing query at Line 1
Commands completed successfully.
1:00:07 PM Started executing query at Line 3
Commands completed successfully.
Total execution time: 00:00:00.030

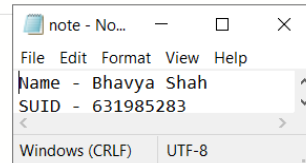
```

6. Execute the stored procedure in step 5 to demonstrate the rollback works. You should give a 6 star rating and then execute again where someone attempts to rate themselves. Produce as screen shot as evidence the rollback worked.

```
1  exec p_rate_user @rating_by_user_id=1, @rating_for_user_id=4, @rating_astype='Buyer', @rating_value=5, @rating_comment='Yellow'
```

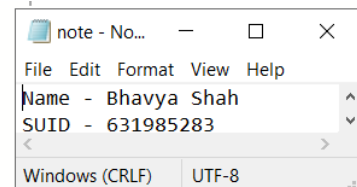
Messages

```
1:01:24 PM  Started executing query at Line 1
              (1 row affected)
              Total execution time: 00:00:00.037
```



7. There is a conceptual data requirement which says that no **TinyU** major can have more than 15 students in it. (I know, this seems silly but think of the bigger problem – how to we enforce a specific minimum or maximum cardinality instead of just 1 or “many”?) Write data logic using an instead of trigger to do this.

```
1  use tinyu
2  go
3  drop trigger if exists t_students_max_major_code_count_insteadof
4  go
5  create trigger t_students_max_major_code_count_insteadof
6  on students
7  instead of update, insert as
8  begin
9  declare @count_student_major_id int= (Select count ('student_major_id') as count_student_major_id)
10 if @count_student_major_id>15
11 begin
12 ;
13 throw 50004, 'No changes permitted',1
14 rollback
15 end
16 else begin
17 update students set students.student_gpa=inserted.student_gpa,students.student_major_id=inserted.student_major_id
18 from inserted
19 where students.student_firstname= inserted.student_firstname
20 and students.student_lastname= inserted.student_lastname
21 end
22 end
23 go
```



Messages

```
1:01:24 PM  Started executing query at Line 1
              (1 row affected)
              Total execution time: 00:00:00.037
```

8. Test step 7 by trying to add or update a student and change their major to ADS. The ADS major has 15 students already. Your code should drop/create the trigger and also test the success and failure of the trigger.

Run Cancel Disconnect Change Connection tinyu Estimated Plan Enable Actual Plan Enable SQLCMD Export as Notebook

```

1 update students
2 set student_gpa=3.4 , student_major_id=2
3 where student_firstname='Lilly' and student_lastname='Padz'
4 select * from students where student_firstname='Lilly' and student_lastname='Padz'
5 select * from majors where major_id=2

```

Results Messages

	student_id	student_firstname	student_lastname	student_year_name	student_major_id	student_gpa	student_notes	student_active	student_inactive_da
1	20	Lilly	Padz	Senior	2	3.400	meet with student	NULL	NULL

	major_id	major_code	major_name
1	2	ADS	Applied Data Science

note - No... File Edit Format View Help
 Name - Bhavya Shah
 SUID - 631985283
 Windows (CRLF) UTF-8

Reflection

Use this section to reflect on your learning. To achieve the highest grade on the assignment you must be as descriptive and personal as possible with your reflection.

- What are the key things you learned through the process of completing this assignment?
 Ans – Stored procedures and triggers.
- What were the challenges or roadblocks (if any) you encountered on the way to completing it?
 Ans – Solving different and multiple errors in queries.
- Were you prepared for this assignment? What can you do to be better prepared?
 Ans – No, I was not well prepared. I need to understand this topic more in detail to be better prepared.
- Now that you have completed the assignment rate your comfort level with this week's material. This should be an honest assessment: (choose one)
 2 ==> I somewhat understand the material but sometimes need guidance from others.