Unit 10 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.

-

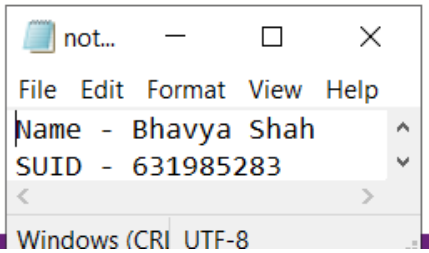
Questions

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.

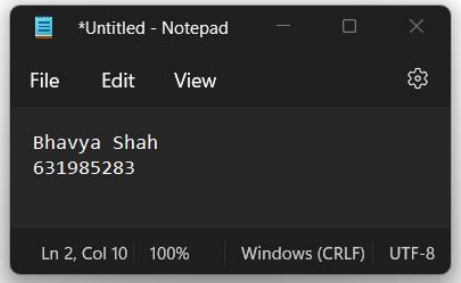
Write the following as SQL programs. For each, include the SQL as a screenshot with the output of the query.

1. In the **TinyU** database,
    a. Write an SQL Stored procedure called **p_upsert_major** which given a major_code (business key) and a major_name does an Upsert, which is the following:
        i. Check if the major_code exists in the table already.
        ii. If yes, update the table and make the major_name match the new major name.
        iii. If no, insert the new major_name and major_code into the table. HINT: major_id is not a surrogate key so you will need to determine the next ID yourself in code!
    b. Test your stored procedure by executing it to make these changes
        i. change : CSC – Computer Sciences to CSC – Computer Science and
        ii. add: FIN – Finance.

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




```
create PROCEDURE p_upsert_major (
    @new_major_code varchar(3),
    @new_major_name varchar(20),
    @new_major_id INT
 ) as BEGIN
if exists (select major_code from dbo.majors where major_code = @new_major_code)
BEGIN
update dbo.majors
set major_name=@new_major_name where major_code=@new_major_code
return NULL
end
ELSE
BEGIN
print 'no major found'
insert into dbo.majors (major_id,major_code,major_name)
values (@new_major_id,@new_major_code,@new_major_name)
return NULL
END
END
exec p_upsert_major @new_major_code = 'CSC' ,@new_major_name = "Computer Science" ,@new_major_id='4'

select * from dbo.majors
```

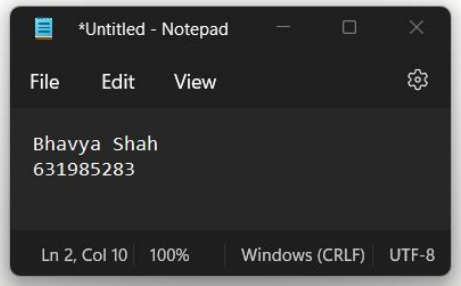


Results Messages

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

Make sure your screenshot captures all up/down code in 1.a AND another screen shot captures 1.b the output of your code execution to show that it works. SELECT the table before and after!

```
create PROCEDURE p_upsert_major (
    @new_major_code varchar(3),
    @new_major_name varchar(20),
    @new_major_id INT
 ) as BEGIN
if exists (select major_code from dbo.majors where major_code = @new_major_code)
BEGIN
update dbo.majors
set major_name=@new_major_name where major_code=@new_major_code
return NULL
end
ELSE
BEGIN
print 'no major found'
insert into dbo.majors (major_id,major_code,major_name)
values (@new_major_id,@new_major_code,@new_major_name)
return NULL
END
END
exec p_upsert_major @new_major_code = 'CSC' ,@new_major_name = "Computer Science" ,@new_major_id='4'
```



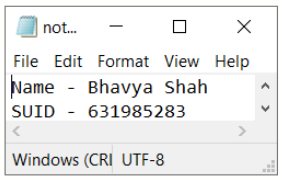
Results | Messages

	major_id	major_code	major_name
1	1	IMT	Information Management and T…
2	2	ADS	Applied Data Science
3	3	ACC	Accounting
4	4	CSC	Computer Science
5	5	BSK	Basket Weaving
6	6	FIN	FINANCE

2. In the **TinyU** database,
    a. write a user-defined function called **f_concat** which combines the any two varchars @a and @b together with a one-character @sep in between.
    For example:
    ```
    select dbo.f_concat('half','baked','-') -- 'half-baked'
    select dbo.f_concat('mike', 'fudge', ' ') -- 'mike fudge'
    ```
    b. Now create a view called **v_students** which displays the student_id student name (first last), student name (last, first), gpa, and name of major. You should call the function you created in 2.a. After you create the view, execute it with a SELECT statement.

Make sure your screenshot captures all up/down code in 2.a AND another screen shot captures 2.b along with the output of the SELECT statement on the view (first few rows is fine).

```
drop function if exists f_concat
go
create function f_concat (
    @a VARCHAR(50),
    @b VARCHAR(50)

) returns varchar(50) as BEGIN
return (@a+ '-' + @b)
end
go
drop view if exists v_students
go
create view v_students as
select student_id,(select dbo.f_concat(student_firstname,student_lastname)) as firstlast_concat, (select dbo.f_concat(student_lastname,student_firstname))
as lastfirst_concat,student_gpa,m.major_name
from dbo.students
join majors m on m.major_id=students.student_major_id
GO
select * from dbo.students
SELECT * from v_students
```



Results Messages

	student_id	firstlast_concat	lastfirst_concat	student_gpa	major_name
1	1	Robin-Banks	Banks-Robin	4.000	Accounting
2	2	Victor-Edance	Edance-Victor	2.404	Applied Data Science
3	3	Erin-Yortires	Yortires-Erin	2.401	Information Management and T…
4	4	Aurora-Borealis	Borealis-Aurora	3.024	Information Management and T…
5	5	Tuck-Androll	Androll-Tuck	3.333	Applied Data Science
6	6	Eura-Quittin	Quittin-Eura	3.372	Applied Data Science
7	7	Willie-Survive	Survive-Willie	2.608	Applied Data Science
8	8	Lola-Dabridgeda	Dabridgeda-Lola	2.732	Information Management and T…
9	9	Doris-Closed	Closed-Doris	3.173	Accounting
10	10	Phil-McCup	McCup-Phil	2.705	Applied Data Science
11	11	Jack-Itupp	Itupp-Jack	3.386	Accounting
12	12	Val-Idation	Idation-Val	3.500	Information Management and T…
13	13	Ida-Knowe	Knowe-Ida	2.724	Computer Sciences
14	14	Lee-Hvmeehom	Hvmeehom-Lee	1.916	Applied Data Science
15	15	Ginger-Beer	Beer-Ginger	4.000	Applied Data Science
16	16	Buck-Naked	Naked-Buck	2.434	Applied Data Science
17	17	Val-Uation	Uation-Val	3.384	Computer Sciences
18	18	Robin-Eue	Eue-Robin	3.006	Applied Data Science
19	19	Tera-Dactyl	Dactyl-Tera	2.367	Information Management and T…

3. In the **TinyU** database,
    a. Write a query on the **majors** table so that the major_name is broken up into keywords one per row. HINT: you must use string_split() with cross apply.

major_id	major_code	major_name	keyword
1	IMT	Information Management and T…	Information
1	IMT	Information Management and T…	Management
1	IMT	Information Management and T…	and
1	IMT	Information Management and T…	Technology

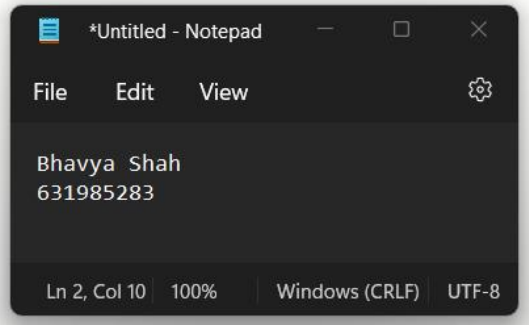
    b. Then use the query in 3.a to create a table-valued function **f_search_majors** which allows you to search the majors by keyword. Demonstrate calling the TVF by querying all majors with the ‘Science’ keyword.

Your screenshot should include the query in 3.a Another screenshot should show the TVF in 3.b and the sample output from the SELECT statement calling the TVF.

```
79  select * from dbo.majors cross apply string_split(major_name,' ')
80  go
81  drop function if exists f_search_majors
82  go
83  create function f_search_majors (
84      @search varchar(50)
85  ) RETURNS TABLE
86  AS
87  return
88  select * from (select major_id, major_code,major_name, value search from majors cross apply string_split(major_name, ' ')) sear
89  where sear.search=@search
90  GO
91  select * from f_search_majors('science')
92
```

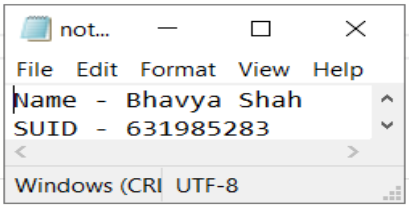
Results | Messages

	major_id	major_code	major_name	search
1	2	ADS	Applied Data Science	Science
2	4	CSC	Computer Science	Science



4. In the **TinyU** database,
    a. Alter the **students** table and add the following columns:
        i. student_active char(1) default ('Y') not null
        ii. student_inactive_date date null

```
1   alter table students
2   add student_active char(1) , student_inactive_data date null
3   GO
4   if exists (select * from INFORMATION_SCHEMA. TABLE_CONSTRAINTS
5       where constraint_name='df_students_student_active')
6       alter table students drop constraint df_students_student_active
7   go
8   alter table students
9   add CONSTRAINT df_students_student_active
10  default 'Y' for student_active
11  go
```

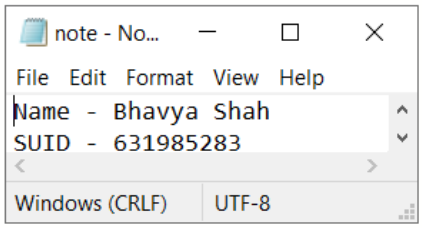


Messages

```
1:09:40 PM      Started executing query at Line 1
                Commands completed successfully.
1:09:40 PM      Started executing query at Line 4
                Commands completed successfully.
1:09:40 PM      Started executing query at Line 8
                Commands completed successfully.
                Total execution time: 00:00:00.145
```

b. Create a trigger on the **students** table which when there is an student_inactive_date set will set student_active to ‘N’, whenever there is not a student_inactive_date then student_active is set to ‘Y’.

```
21  drop trigger if exists t_after_update_student_status
22  go
23  CREATE trigger t_after_update_student_status
24      on students
25      after insert, update
26  as begin
27      if update(student_inactive_data) BEGIN
28          update students
29          set student_active = case
30          when student_inactive_data is not null then 'N'
31          when student_inactive_data is null then  'Y'
32          end
33      END
34  END
```

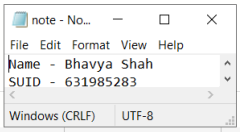


Messages

```
1:18:41 PM     Started executing query at Line 20
               Commands completed successfully.
1:18:41 PM     Started executing query at Line 23
               Commands completed successfully.
               Total execution time: 00:00:00.037
```

c. Write SQL code to deactivate all the ‘Graduate’ students with a date of ‘2020-08-01’

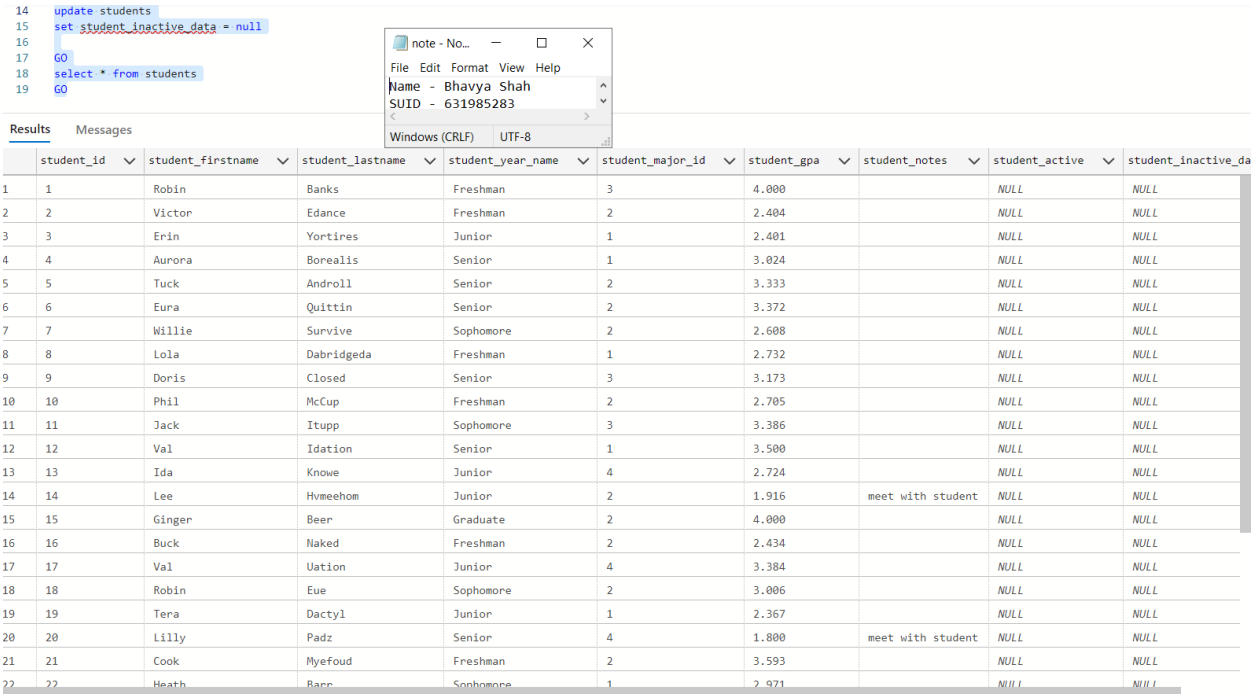
```
update students
set student_inactive_data = '2020-08-01'

GO
select * from students
GO
```



Results Messages

	student_id	student_firstname	student_lastname	student_year_name	student_major_id	student_gpa	student_notes	student_active	student_inactiv
1	1	Robin	Banks	Freshman	3	4.000		NULL	2020-08-01
2	2	Victor	Edance	Freshman	2	2.404		NULL	2020-08-01
3	3	Erin	Yortires	Junior	1	2.401		NULL	2020-08-01
4	4	Aurora	Borealis	Senior	1	3.024		NULL	2020-08-01
5	5	Tuck	Androll	Senior	2	3.333		NULL	2020-08-01
6	6	Eura	Quittin	Senior	2	3.372		NULL	2020-08-01
7	7	Willie	Survive	Sophomore	2	2.608		NULL	2020-08-01
8	8	Lola	Dabridgeda	Freshman	1	2.732		NULL	2020-08-01
9	9	Doris	Closed	Senior	3	3.173		NULL	2020-08-01
10	10	Phil	McCup	Freshman	2	2.705		NULL	2020-08-01
11	11	Jack	Itupp	Sophomore	3	3.386		NULL	2020-08-01
12	12	Val	Idation	Senior	1	3.500		NULL	2020-08-01
13	13	Ida	Knowe	Junior	4	2.724		NULL	2020-08-01
14	14	Lee	Hvmeehom	Junior	2	1.916	meet with student	NULL	2020-08-01
15	15	Ginger	Beer	Graduate	2	4.000		NULL	2020-08-01
16	16	Buck	Naked	Freshman	2	2.434		NULL	2020-08-01
17	17	Val	Uation	Junior	4	3.384		NULL	2020-08-01
18	18	Robin	Eue	Sophomore	2	3.006		NULL	2020-08-01
19	19	Tera	Dactyl	Junior	1	2.367		NULL	2020-08-01
20	20	Lilly	Padz	Senior	4	1.800	meet with student	NULL	2020-08-01
21	21	Cook	Myefoud	Freshman	2	3.593		NULL	2020-08-01

d. Write SQL code to re-activate all the ‘Graduate’ students.



```
14  update students
15  set student_inactive_data = null
16
17  GO
18  select * from students
19  GO
```

note - No...

File Edit Format View Help

Name - Bhavya Shah
SUID - 631985283

Windows (CRLF) UTF-8

Results Messages

	student_id	student_firstname	student_lastname	student_year_name	student_major_id	student_gpa	student_notes	student_active	student_inactive_da
1	1	Robin	Banks	Freshman	3	4.000		NULL	NULL
2	2	Victor	Edance	Freshman	2	2.404		NULL	NULL
3	3	Erin	Yortires	Junior	1	2.401		NULL	NULL
4	4	Aurora	Borealis	Senior	1	3.024		NULL	NULL
5	5	Tuck	Androll	Senior	2	3.333		NULL	NULL
6	6	Eura	Quittin	Senior	2	3.372		NULL	NULL
7	7	Willie	Survive	Sophomore	2	2.608		NULL	NULL
8	8	Lola	Dabridgeda	Freshman	1	2.732		NULL	NULL
9	9	Doris	Closed	Senior	3	3.173		NULL	NULL
10	10	Phil	McCup	Freshman	2	2.705		NULL	NULL
11	11	Jack	Itupp	Sophomore	3	3.386		NULL	NULL
12	12	Val	Idation	Senior	1	3.500		NULL	NULL
13	13	Ida	Knowe	Junior	4	2.724		NULL	NULL
14	14	Lee	Hvmeehom	Junior	2	1.916	meet with student	NULL	NULL
15	15	Ginger	Beer	Graduate	2	4.000		NULL	NULL
16	16	Buck	Naked	Freshman	2	2.434		NULL	NULL
17	17	Val	Uation	Junior	4	3.384		NULL	NULL
18	18	Robin	Eue	Sophomore	2	3.006		NULL	NULL
19	19	Tera	Dactyl	Junior	1	2.367		NULL	NULL
20	20	Lilly	Padz	Senior	4	1.800	meet with student	NULL	NULL
21	21	Cook	Myefoud	Freshman	2	3.593		NULL	NULL
22	22	Heath	Barr	Sophomore	1	2.971		NULL	NULL

Provide a screenshot of your code from 4.a. and 4.b working. Provide another screenshot demonstrating 4.c worked. Then a final screenshot of code and demonstration of 4.d working.

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.

1. What are the key things you learned through the process of completing this assignment?
   Ans – Triggers and procedures

2. What were the challenges or roadblocks (if any) you encountered on the way to completing it?
   Ans - It took a lot of time to understand the concept.

3. Were you prepared for this assignment? What can you do to be better prepared?
   Ans - No, I was not prepared. I need to learn and practice more.

4. Now that you have completed the assignment rate your comfort level with this week's material. This should be an honest assessment: (choose one)

   4 ==> I understand this material and can explain it to others.
   3 ==> I understand this material.
   2 ==> I somewhat understand the material but sometimes need guidance from others.
   1 ==> I understand very little of this material and need extra help.