Unit 10 Problem Set Submission Form

Overview

Your Name	Jainish Savaliya
Your SU Email	jsavaliy@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.

-

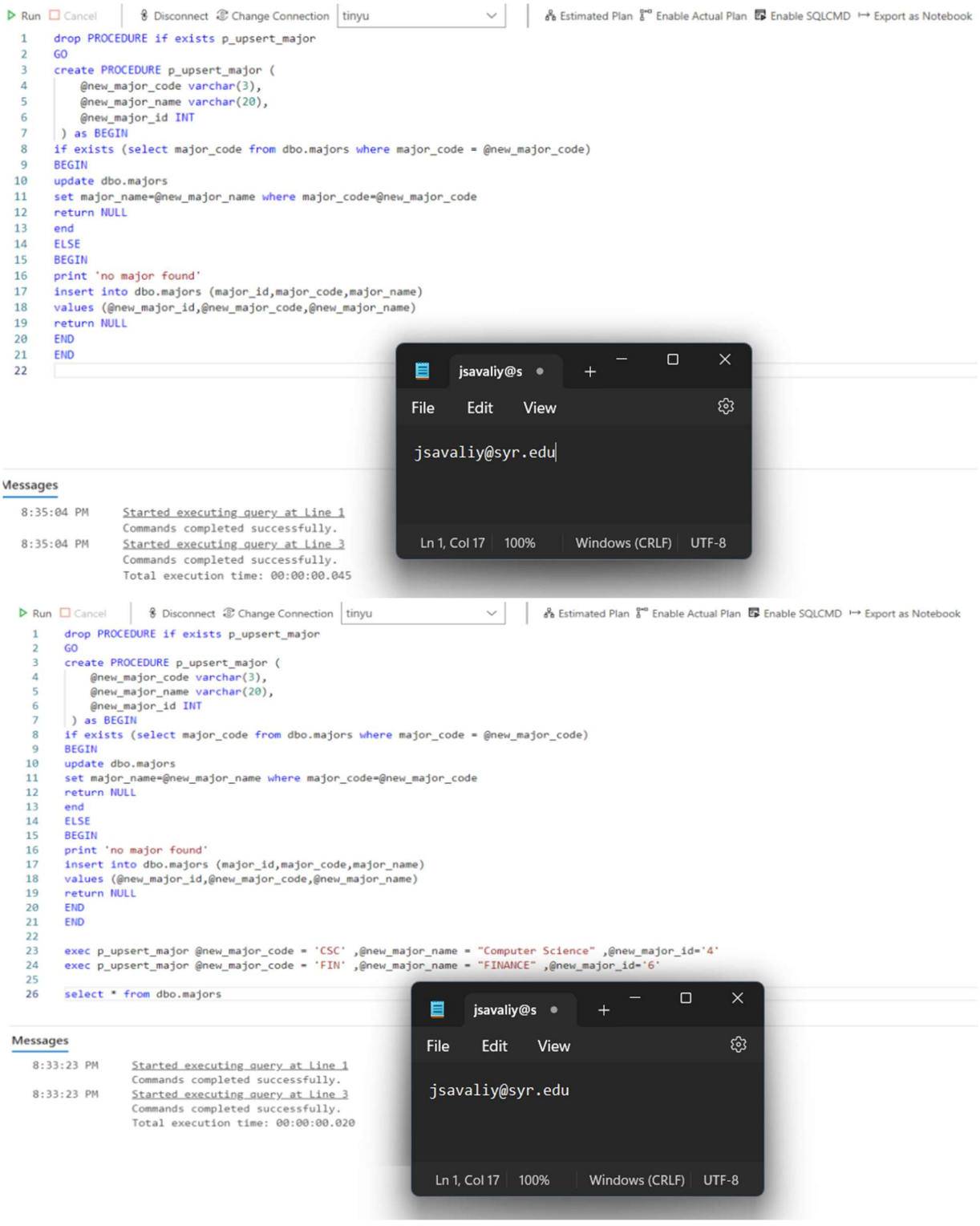
Questions

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.

    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!



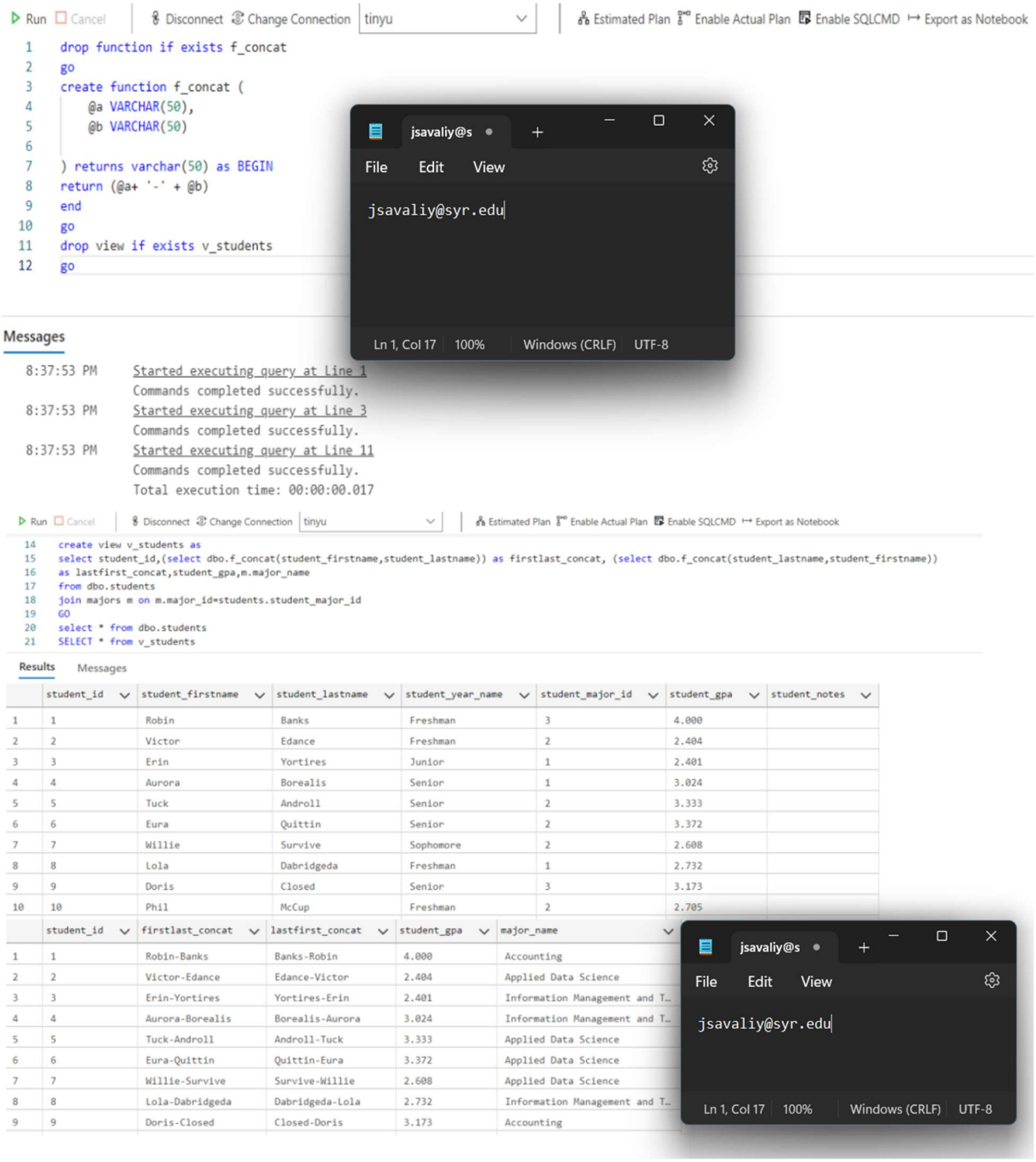
2. In the **TinyU** database,
    a. write a user-defined function called **f_concat** which combines the any two varchars @a and @b together with a one-character @sep in between.

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).

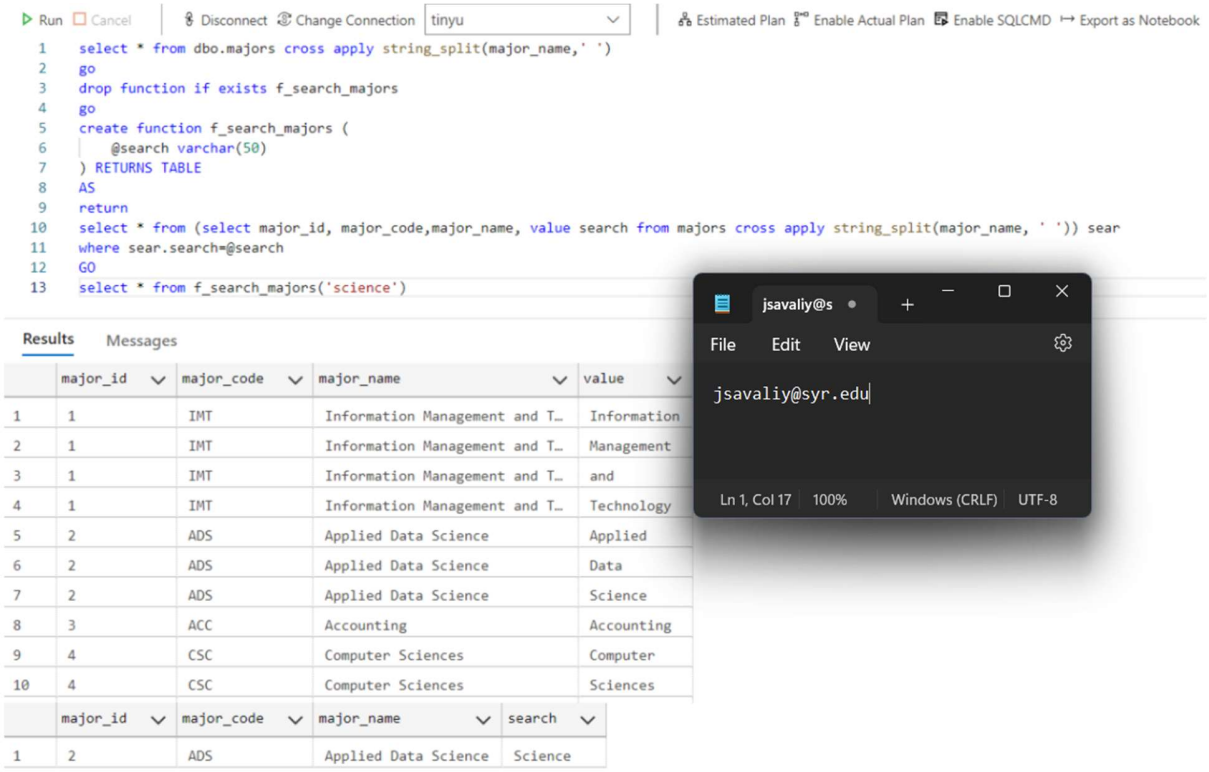


3. In the **TinyU** database,
    a. Write a query on the **majors** table so that the major_name is broken up into keywords one per row. HINT: you must use string_split() with cross apply.

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.

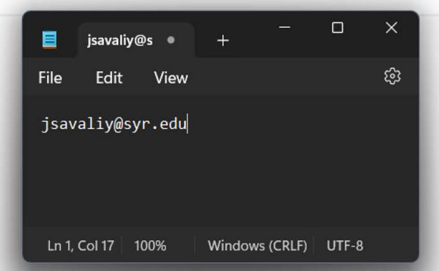


4. In the **TinyU** database,
    a. Alter the **students** table and add the following columns:
        i. student_active char(1) default ('Y') not null
        ii. student_inactive_date date null
    b. Create a trigger on the **students** table which when there is an student_inactive_date set will set student_active to 'N', whenever there is not a student_inactive_date then student_active is set to 'Y'.
    c. Write SQL code to deactivate all the 'Graduate' students with a date of '2020-08-01'
    d. Write SQL code to re-activate all the 'Graduate' students.

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.

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

```
drop trigger if exists t_after_update_student_status
go
CREATE trigger t_after_update_student_status
    on students
    after insert, update
as BEGIN
    if update(student_inactive_data) BEGIN
        update students
        set student_active = case
        when student_inactive_data is not null then 'N'
        when student_inactive_data is null then 'Y'
        END
    END
END
```



Messages

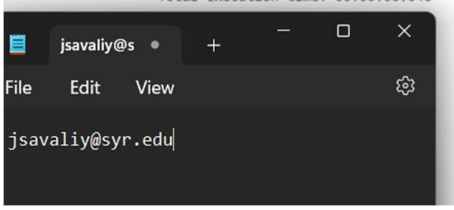
```
9:12:45 PM    Started executing query at Line 1
              Commands completed successfully.
9:12:45 PM    Started executing query at Line 3
              Commands completed successfully.
              Total execution time: 00:00:00.016
```

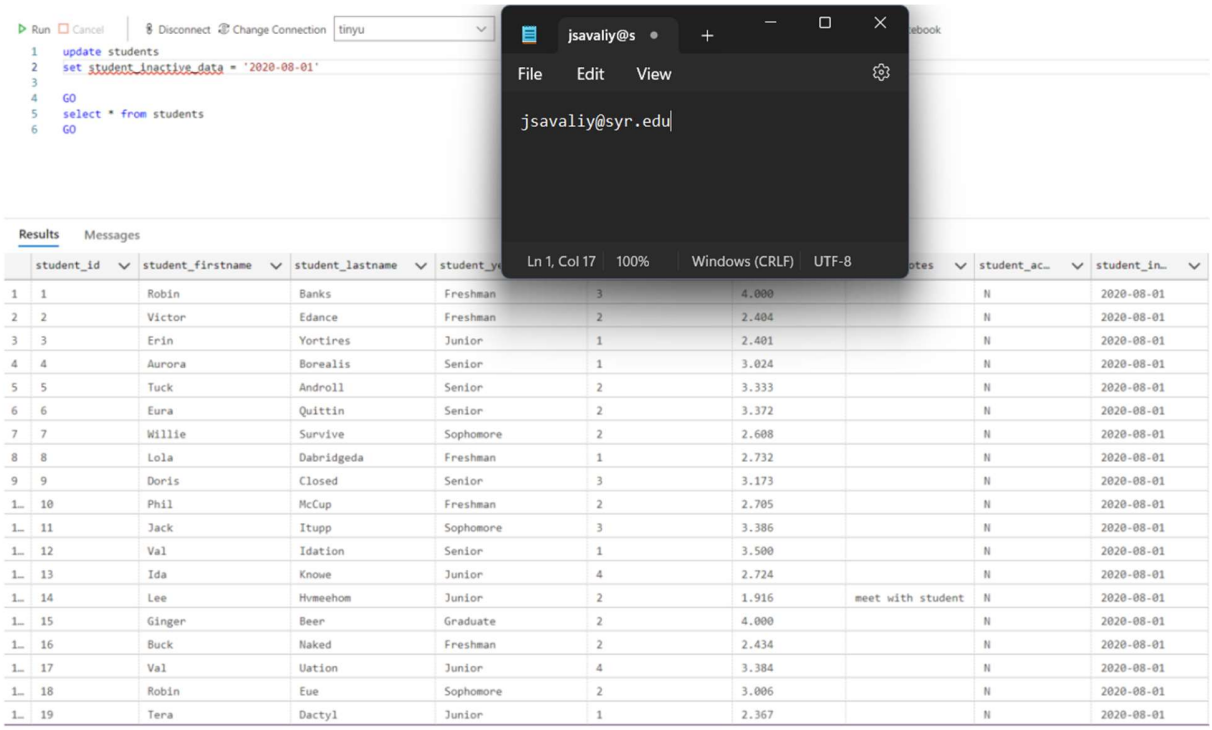
Run | Cancel | Disconnect | Change Connection | tinyu | Estimated Plan | Enable Actual Plan | Enable SQLCMD | Export as Notebook

```
alter table students
add student_active char(1) , student_inactive_data date null
GO
if exists (select * from INFORMATION_SCHEMA. TABLE_CONSTRAINTS
    where constraint_name='df_students_student_active')
    alter table students drop constraint df_students_student_active
go
alter table students
add CONSTRAINT df_students_student_active
default 'Y' for student_active
go
```

Messages

```
9:11:01 PM    Started executing query at Line 1
              Commands completed successfully.
9:11:01 PM    Started executing query at Line 4
              Commands completed successfully.
9:11:01 PM    Started executing query at Line 8
              Commands completed successfully.
              Total execution time: 00:00:00.046
```





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

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?
   -> The key things I would say I learnt in this assignment are views, Triggers and stored procedures.

2. What were the challenges or roadblocks (if any) you encountered on the way to completing it?
   -> I watched the video and understood the concepts so this one was smooth compared to other assignments

3. Were you prepared for this assignment? What can you do to be better prepared?
   -> I think I was prepared for this assignment. The better I can do is to dig deep into the concepts and try solving new problems of this topics.

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.