DTSC660: Data and Database Management with SQL
Module 5
Assignment 5

Directions:

You will have ***one attempt*** at submitting this assignment. The assignment will be manually graded by the course grader. Read all directions. Be sure to name your functions, procedures, and triggers ***exactly*** as indicated. You must submit your solutions in the .sql file template that I have supplied. This file includes comments that delineate the file into sections - one section per problem. Be sure to write your name and at the top of the .sql file at the location I have indicated in the template. You must rename the .sql file according to the following format prior to submission:

```
<LastName>_<DOB_DayOfMonth>_Assignment_5.sql
```

Where `<LastName>` is your last name and `<DOB_DayOfMonth>` is the day of the month you were born. This naming convention is merely meant to make the grading process easier. For example, my .sql file would have the name:

```
Hollers_21_Assignment_5.sql
```

Submissions that do not adhere to these guidelines may receive a grade of zero.

Question 1:

Write a SQL function that accepts a principal mortgage amount, an annual percentage rate (APR), and the number of years a mortgage will be paid back over. Calculate the associated monthly mortgage payment according to the following annuity formula:

$$A = P\left(i + \frac{i}{(1+i)^{n} - 1}\right)$$

where:

A = Monthly Payment Amount
P = Principle (Initial) Mortgage Amount
i = APR / 12 = Monthly Interest Rate
n = years * 12 = Total Number of Payments

Be sure to specify that data types of the input arguments have enough significant digits to account for realistic mortgage amounts (as large as in the millions of dollars) and APRs (down to the sixth decimal place will be sufficient).

There are multiple people in the class with the same last name, so the following naming convention must be followed exactly. You must name your function:

```
<LastName>_<DOB_DayOfMonth>_monthlyPayment
```

Where `<LastName>` is your last name and `<DOB_DayOfMonth>` is the day of the month you were born. This naming convention is merely meant to make the grading process easier, so that all of the functions, procedures, and triggers have unique names.

How would I invoke this function? Below is an example of my function invocation (call), where I am calculating the monthly payment for a mortgage with a principal amount of $250,000, an APR of 4.125%, and a loan duration of 30 years:

```
SELECT Hollers_21_monthlyPayment(250000.00, 0.04125, 30);
```

Be sure to follow the same format that I have used for specifying the input parameters; namely, an APR of 4.125% should be specified as `0.04125`.

Make sure you include an invocation that uses the above stated parameters in your solution.

Question 2:

For this problem, the registrar's office has requested that we create a report of students and the number of courses they have taken to verify minimum graduation requirements. You have been tasked with creating a table and procedure that track this information. To complete this problem:

a. create a table (NOT a temporary table) called *student_course_nums* within the university database schema.

b. Write a procedure that accepts a student ID as input. You must name your procedure:

```
<LastName>_<DOB_DayOfMonth>_stuCourseNumsProc
```

Where `<LastName>` is your last name and `<DOB_DayOfMonth>` is the day of the month you were born. Below is an example of how I named my procedure:

```
Hollers_21_stuCourseNumsProc
```

c. The procedure calculates the total number of course sections attended by that student, and adds a tuple to the *student_course_nums* table consisting of the student's ID number, name, and total courses attended - call these attributes: ID, name, and tot_courses.

d. If the student already has an entry in the table, then the procedure makes sure the total number of courses attended in the *student_course_nums* table is up-to-date.
e. Make sure in your solution that you call the procedure.