

---

## CSC 352/452: DATABASE PROGRAMMING

### ASSIGNMENT # 2 (100 Points) **Due Date 7/30 11:59pm**

---

#### SECTION 1: PL/SQL Fundamentals

**Q1 (10 Points)** For each declaration below, indicate whether it is legal or not. Explain if it is not legal.

```
v_empno          NUMBER(4);
v_a, v_b, v_c     VARCHAR2(10);
v_address         VARCHAR2(30) NOT NULL;
v_in_stock        BOOLEAN := 1;
```

Save your answers in the file [Section1Q1](#).

**Q2 (10 Points)** Your first PL/SQL program - Create an anonymous PL/SQL block to output the phrase “My PL/SQL Program Works” to the screen. Save your code in the file [Section1Q2.sql](#).

**Q3 (15 Points)** Create an anonymous PL/SQL block that accepts the current salary and raise percentage through SQL\*Plus substitution variables and calculate the new salary based on the input:

$$\text{new salary} := \text{old salary} * (1 + \text{raise percentage} / 100)$$

The result should be stored in a PL/SQL variable and printed on the screen.

Your program should handle NULL values. A NULL value entered for either one or both of the input values is equivalent to a numerical 0 (Note: To associate NULL values for your SQL\*Plus substitution variables, just enter NULL.)

Save your code in the file [Section1Q3.sql](#).

**Q4 (15 Points)** Create an anonymous PL/SQL block that accepts an integer number  $N$  through SQL\*Plus substitution variable and then determines for each of the numbers in the range 1 through  $N$  inclusive whether it is odd or even. Use the MOD function to determine whether a number is odd or even. For example,  $\text{MOD}(10,2) = 0$  and  $\text{MOD}(11,2) = 1$ . Print the results on the screen.

Your program should handle NULL values.  $N$  should be set to 0 if a NULL value is entered.

Save your code in the file [Section1Q4.sql](#).

---

#### SECTION 2: Exceptions and Sub-programs

**Q1 (50 Points)** Write a PL/SQL procedure, *parse\_name*, which accepts a string representing names and returns the first name, the last name, and the title. The first name and last name returned from the procedure should be in upper cases while the title returned should preserve the original case. The input name string is in one of the following two formats:

```
FIRST_NAME LAST_NAME TITLE
LAST_NAME, FIRST_NAME TITLE
```

The first format depicts that the name string starts with a person's first name and last name with a space in between, followed by a space, and then the title of the person. On the other hand, the second format states that the name string starts with a person's last name, followed by a comma and a space, and then the first name suffixed with the title with a space before it.

The following are examples of valid name strings conforming to the formats:

- 1) Jane Doe Ms.
- 2) Doe, Jane Ms.

Your procedure should return JANE for first name, DOE for last name, and Ms. for title for the above two input strings.

Also, the following are examples of valid name strings conforming to the formats:

- 1) Jane Doe MS.
- 2) Doe, Jane MS.

Your procedure should return JANE for first name, DOE for last name, and MS. for title for the above two input strings.

Your procedure should generate exceptions if any one of the three components of the input name string is missing. Further, a message should be printed indicating that the input string is invalid. For instance, the following input strings should cause an exception to be generated and the appropriate messages printed:

- 1) Doe
- 2) Jane Doe
- 3) Doe, Jane

You should wrap *parse\_name* within an anonymous PL/SQL block that accepts the name string through SQL\*Plus substitution variables, calls *parse\_name* with the input string, and then prints the first name, last name, and title returned from the *parse\_name* procedure. You should print the title first, followed by the first name and then the last name, separated by spaces.

Save your work in the file named **Section2Q1.sql**.

---

**Assignment Instructions:** This assignment consists of two sections. Section 1 is divided into four parts, each of which may require you to submit a file. Section 2 has one part, which may also require you to submit a file.

To complete this assignment, please follow these steps:

1. Create a folder named "LastName HW2" in your home directory (e.g., c:\LastName HW2).
2. Save the solution files for each section/question in the appropriate subfolders within the "LastName HW2" folder. For Section 1, save "Section1Q1.doc" and the script files "Section1Q2.sql," "Section1Q3.sql," and "Section1Q4.sql" in the "Section1" subfolder. For Section 2, save the script file "Section2Q1.sql" in the "Section2" subfolder.
3. Once you have saved all the files, compress the entire "LastName HW2" folder into a zip file.
4. Rename the zip file as "LastName HW2.zip".
5. Finally, submit the zipped file to the Assignment 2 Submission page on D2L/your course website.

Please make sure to submit your assignment in the designated submission location.

For example, if your last name is Smith and this is assignment #2, your folder structure and file placement should be as follows:

**Folder: c:\Smith HW2**

- Subfolder: Section1
  - Section1Q1.doc
  - Section1Q2.sql
  - Section1Q3.sql
  - Section1Q4.sql
- Subfolder: Section2
  - Section2Q1.sql

After creating the folder structure and saving the files, you should compress the "LastName HW2" folder into a zip file named "LastName HW2.zip" and submit it to the Assignment 2 Submission link on D2L.

If you have any further questions, please let me know.

**SUBMIT YOUR HW2 FOLDER AS ZIP FILE TO YOUR D2L ASSIGNMENT 2 SUBMISSION LINK FOR GRADING. Make sure only one copy is submitted.**