

COMP40725

Lab Book 7

- Create a folder called **Lab7** in your **COMP40725** folder. You may wish to have this inside a folder that is being synced by Google Drive.
- Use Notepad++ or TextWrangler to write SQL queries. Test these queries in your **Oracle 11g XE Database** system. You should use **SQL Command Line (sqlplus tool)**.
- Use comments in your sql queries: `--` is for a single line, and `/*...*/` for multiline.
- To enable output to screen on **sqlplus** run the following command when you start your session. It must be set for each user and only lasts for the session: **SET SERVEROUTPUT ON;**
- Use the command **show errors;** to illustrate errors

Hint: This lab requires you to apply what was covered in **Lecture 12**, with some extra (existing) functions. Some samples of using these functions are given below. Before starting the lab, it is recommended that you run the following queries to see how these functions work.

```
SELECT 'Firstname Surname' FROM DUAL;
SELECT TRIM('Firstname Surname') FROM DUAL;
SELECT UPPER('Firstname Surname') FROM DUAL;
SELECT LOWER('FIRSTNAME Surname') FROM DUAL;
SELECT INITCAP('FIRSTNAME SURNAME') FROM DUAL;
SELECT VSIZE('Firstname Surname') FROM DUAL;
SELECT REPLACE('This is an example', 'his', 'hat') FROM DUAL;
```

1. Copy the PL/SQL Hello World sample **procedure** and **function** from the **lecture 12** notes.
2. Create a procedure that has two inputs - Firstname and Surname. The procedure should DBMS output the following 3 lines:

firstname surname (lowercase)

FIRSTNAME SURNAME (uppercase)

Firstname Surname (Initial uppercase)

The following PL/SQL functions can be used to assist:

UPPER(VARCHAR2), LOWER(VARCHAR2), INITCAP(VARCHAR2)

3. Create a procedure that has one input, a person's full name. The procedure should DBMS output the following lines:

Firstname Surname (initial uppercase - with any space characters before or after the person's name removed)

Number of bytes in the internal representation of the person's name

The following PL/SQL functions can be used to assist:

INITCAP(VARCHAR2), TRIM(VARCHAR2), VSIZE(expression)

4. Create a procedure that removes all vowels from a string. Supply an anonymous PL/SQL block that demonstrates the procedure and uses the DBMS output put_line procedure to output it. (Hint: remember to declare the procedure parameter as "IN OUT")

The following PL/SQL functions can be used to assist:

TRIM(VARCHAR2), REPLACE

5. Create a function that inputs a number and returns: number*2
6. Create a function that inputs 5 numbers and returns the sum of them all minus 1 (i.e. (1+2+3+4+5)-1)
7. Overloading: Create a package that contains 4 functions with the same name:
 - I. The first function should have 2 number parameters, and return the sum - 1
 - II. The second function should have 3 number parameters, and return the sum - 1
 - III. The third function should have 4 number parameters, and return the sum - 1
 - IV. The fourth function should have 5 number parameters, and return the sum - 1.

Demonstrate the use of the functions from an anonymous PL/SQL block (outside of the package) by using DBMS_OUTPUT.PUT_LINE

Submit the SQL statements in a “.sql” that you used to test all of the above queries and name it “Lab7_StudentNumber_FirstName_LastName.sql**”.**