**CSCI N311**

## Project 8: PL/SQL subprograms.

For this exercise, you will create a table and populate it with hundreds of unique rows of test data. These rows of data will based upon 10 rows of data in a table called BBT\_USERS\_TEMP. You will create this table by copying any10 rows from the table you created in project 5 (problem 2) using “create table from an existing table” technique.

***Step 1 – Create the BBT\_USERS\_TEMP table(10’).***

### *Step 2 - Create PHONE\_USERS table (10’)*

Create a PHONE\_USERS table to store the results of your procedure. The table is defined as follows:

TELEPHONE\_NUMBER NOT NULL VARCHAR2(80)

FIRST\_NAME VARCHAR2(80)

LAST\_NAME VARCHAR2(80)

KEYMAP\_LASTNAME CHAR(4)

PASSWORD VARCHAR2(80)

The column TELEPHONE\_NUMBER should be defined as the primary key. Also, create a unique constraint on LAST\_NAME.

### *Step 3 - create function(20’)*

Function name: KEYMAP  
Input Parameter: VARCHAR2, any string which will usually be a person's last name  
  
Output value: VARCHAR2, a 4-character string which represents their telephone extension. For purposes of this project, the telephone extension will be based on the first four characters of their last name, 2 = ABC, 3 = DEF, 4 = GHI, 5=JKL, 6=MNO, 7=PQRS, 8=TUV, 9=WXYZ just like the letters on a telephone keypad. For names shorter than 4 characters, use zeros in those character positions. The Oracle function TRANSLATE will be helpful.

### *Step 4 - create procedure (35’)*

You are to write a PL/SQL procedure that will iterate n times where n is the parameter you pass into your procedure. For each iteration of your procedure you will copy 10 rows of data from BBT\_USERS\_TEMP table into your PHONE\_USERS table, e.g. if n is 5, after the procedure call, PHONE\_USERS table will have 50 rows of data.

Examples of calling stored procedure:

execute sp\_p(5); -- should result in 50 unique rows in PHONE\_USERS table

delete from PHONE\_USERS;

execute sp\_p(20); -- should result in 200 unique rows in the PHONE\_USERS table  
  
Data inserted into the PHONE\_USERS table must meet the following requirements:

1. Each TELEPHONE\_NUMBER should be unique. Make it is unique any way you want as long as it looks like a telephone number in the format (999) 999-9999. (10’)
2. LAST\_NAME should be LAST\_NAME with 4 digits appended to the end. These digits will be 0001 through 000n where n is the iteration. Assuming n is smaller than 9999. (10’)
3. KEYMAP\_LASTNAME is returned from the function call. It will be the telephone keymap for the 1st four letters of the last name. **Use the function you created in step 3.** (5’)
4. All other columns will be copied exactly as they appear in the BBT\_USERS\_TEMP table. Extra columns in BBT\_USERS\_TEMP should be disregarded. (10’)

### *Step 5 - Create package (20’)*

Create a package which contains both your function and your procedure.

***Step 6 – Run the programs in the package to test. (5’)***

***Submission:***

Submit the sql file.