Department of Computer Science and Engineering

Islamic University of Technology (IUT)

A subsidiary organ of OIC

Lab-6 Report

CSE 4508: RDBMS Lab Report

Name: Anm Zahid Hossain Milkan

Student ID: 200041202

Section: 2B

Semester: 5th

Academic Year: 2022-23

Date of Submission: 17/10/23

Task B: Write a PL/SQL procedure that takes as input a string. The program will achieve two things:

1) Make a new string with a space added between every character of the input string.

2) Check if the original input string was a palindrome. Print “Yes” or “No” accordingly.

[For example: “twat” is not a palindrome, but “tawat” is]

Solution:

```
SET SERVEROUTPUT ON;

VARIABLE input_string VARCHAR2(100);
VARIABLE modified_string VARCHAR2(200);
VARIABLE reversed_string VARCHAR2(100);

-- Set the input string directly
EXEC :input_string := 'tawat';

DECLARE
    i NUMBER;
BEGIN
    :modified_string := '';
    IF :input_string IS NOT NULL THEN
        FOR i IN 1..LENGTH(:input_string) LOOP
            :modified_string := :modified_string ||
SUBSTR(:input_string, i, 1) || ' ';
        END LOOP;
        :modified_string := RTRIM(:modified_string, ' ');
```

```
    END IF;
END;
/

DECLARE
    i NUMBER;
    reversed_input VARCHAR2(100);
BEGIN
    IF :input_string IS NOT NULL THEN
        reversed_input := '';
        FOR i IN REVERSE 1..LENGTH(:input_string) LOOP
            reversed_input := reversed_input ||
SUBSTR(:input_string, i, 1);
        END LOOP;
    ELSE
        reversed_input := NULL;
    END IF;

    IF :input_string = reversed_input THEN
        DBMS_OUTPUT.PUT_LINE('Yes, it is a palindrome.');
    ELSE
        DBMS_OUTPUT.PUT_LINE('No, it is not a
palindrome.');
    END IF;

    DBMS_OUTPUT.PUT_LINE('Modified String: ' ||
:modified_string);
END;
/
```

Output:

```
Yes, it is a palindrome.
Modified String: t a w a t

PL/SQL procedure successfully completed.

SQL>
```

Task C: Write a PL/SQL procedure/function called nearest_primes. This procedure takes a number, n,as an input. Given the number n, the procedure will output the nearest prime number less than n and the nearest prime number greater than n. For example, if n = 15, the program should output 13 and 17.

Solution:

```
CREATE OR REPLACE PROCEDURE nearest_primes(n NUMBER) IS
    less_prime NUMBER;
    greater_prime NUMBER;
    foundL BOOLEAN := FALSE;
    foundG BOOLEAN := FALSE;

    FUNCTION is_prime(num IN NUMBER) RETURN BOOLEAN IS
        is_prime BOOLEAN := TRUE;
    BEGIN
        IF num <= 1 THEN
            is_prime := FALSE;
        ELSIF num <= 3 THEN
            is_prime := TRUE;
        ELSIF num MOD 2 = 0 OR num MOD 3 = 0 THEN
            is_prime := FALSE;
        ELSE
            FOR i IN 5..SQRT(num) LOOP
                IF num MOD i = 0 OR num MOD (i + 2) = 0 THEN
                    is_prime := FALSE;
                    EXIT;
```

```
                END IF;
            END LOOP;
        END IF;
        RETURN is_prime;
    END is_prime;

BEGIN
    IF n <= 1 THEN
        DBMS_OUTPUT.PUT_LINE('No prime numbers less than or greater than '
|| n);
        RETURN;
    END IF;

    less_prime := n - 1;
    greater_prime := n + 1;

    WHILE (NOT foundL OR NOT foundG) AND (less_prime >= 2 OR
greater_prime >= 2) LOOP
        IF is_prime(less_prime) AND NOT foundL THEN
            foundL := TRUE;
        ELSIF foundL THEN
            less_prime := less_prime;
        ELSE
            less_prime := less_prime - 1;
        END IF;

        IF is_prime(greater_prime) AND NOT foundG THEN
            foundG := TRUE;
        ELSIF foundG THEN
            greater_prime := greater_prime;
        ELSE
            greater_prime := greater_prime + 1;
        END IF;

        -- Exit the loop if both conditions are met
        IF foundL AND foundG THEN
            EXIT;
        END IF;
    END LOOP;
```

```
    IF foundG THEN
        DBMS_OUTPUT.PUT_LINE('Nearest prime less than ' || n || ' is ' ||
less_prime);
    END IF;

    IF foundL THEN
        DBMS_OUTPUT.PUT_LINE('Nearest prime greater than ' || n || ' is '
|| greater_prime);
    END IF;

    IF NOT foundL AND NOT foundG THEN
        DBMS_OUTPUT.PUT_LINE('No prime number found in either direction
for ' || n);
    END IF;
END nearest_primes;
/



ACCEPT n_input PROMPT 'Enter the number : ';
DECLARE
   n_input NUMBER;
BEGIN
  n_input := &ninput;
  nearest_primes(n_input);
END;
```

Output:

```
SQL> DECLARE
  2     n_input NUMBER;
  3  BEGIN
  4    n_input := &ninput;
  5    find_nearest_primes(n_input);
  6  END;
  7  /
Enter value for ninput: 29
old   4:   n_input := &ninput;
new   4:   n_input := 29;
Nearest prime less than 29 is 23
Nearest prime greater than 29 is 35
```