Islamic University of Technology

RDBMS

CSE 4508

Lab Report 6

Name : Amina

Student ID : 200041155

Section : 1

Lab Group : 1B (shifted from 1A)

Date of Performance : 18/10/23

Date of Submission : 18/10/23

Task A

In task A, at first I created a table called users with 2 columns Username and password length. Then I wrote a block of PL/SQL to print the maximum password from the table.

Also in that block I printed the number of permutations the hacker have to go through to crack the password. This is done using a for loop which calculates the permutation.

Table creation and PL/SQL block:

```
CREATE TABLE USERS(
    USERNAME VARCHAR2(10),
    PASS_LEN NUMBER
);
```

```
SET SERVEROUTPUT ON;

DECLARE
    ML    NUMBER;
    PERMU NUMBER;
    NUM   NUMBER;
BEGIN
    SELECT
        MAX(PASS_LEN) INTO ML
    FROM
        USERS;
    DBMS_OUTPUT.PUT_LINE(ML
                         || ' is maximum password length');
    PERMU := 1;
    NUM := 52;
    FOR I IN 1..ML LOOP
        PERMU := PERMU * NUM;
        NUM := NUM - 1;
    END LOOP;
    DBMS_OUTPUT.PUT_LINE(PERMU
                         || ' is the permutation number');
END;
/
```

Result:

```
6 is maximum password length
14658134400 is the permutation number
```

USERNAME	PASS_LEN
Doll	3
Doraemon	2
Shinchan	5
Hatori	4
Nobita	3
Dorami	6
Shizuka	4
Karayel	2
Mimi	5

Task B

In this task, we were asked to create a procedure to show nearest prime number greater than the given number and the nearest prime number smaller than the given number.

I created a procedure which takes a number as input and find two nearest prime numbers one less and one greater than the input. This is done using two while loops. One while loop decrements the given number one by one and checks if the number is prime or not. Another does it by incrementing.

Below is the implementation of the procedure :

```
SET SERVEROUTPUT ON;

CREATE OR REPLACE PROCEDURE GENERATE_NEAREST_PRIME(
    S IN NUMBER
) IS
    M   NUMBER;
    P   NUMBER;
    PRI NUMBER;
BEGIN
    M := S;
    PRI := 0;
    WHILE (TRUE) LOOP
        M := M - 1;
        P := 2;
        PRI := 1;
        WHILE P*P <= M LOOP
            IF ( MOD(M, P) = 0 ) THEN
                PRI := 0;
                EXIT;
            ELSE
                P := P + 1;
            END IF;
        END LOOP;
        IF (PRI = 1) THEN
            EXIT;
        END IF;
    END LOOP;
    DBMS_OUTPUT.PUT_LINE(M
                         || ' is nearest prime less than n');
    M := S;
    PRI := 0;
    WHILE (TRUE) LOOP
        M := M + 1;
        P := 2;
        PRI := 1;
        WHILE P*P <= M LOOP
            IF ( MOD(M, P) = 0 ) THEN
                PRI := 0;
                EXIT;
            ELSE
                P := P + 1;
            END IF;
        END LOOP;
        IF (PRI = 1) THEN
            EXIT;
        END IF;
    END LOOP;
    DBMS_OUTPUT.PUT_LINE(M
                         || ' is nearest prime greater than n');
END GENERATE_NEAREST_PRIME;
/
```

Result:

```
SQL> BEGIN
  2      GENERATE_NEAREST_PRIME(15);
  3  END;
  4  /
13 is nearest prime less than n
17 is nearest prime greater than n

PL/SQL procedure successfully completed.

SQL>
SQL> BEGIN
  2      GENERATE_NEAREST_PRIME(10);
  3  END;
  4  /
7 is nearest prime less than n
11 is nearest prime greater than n

PL/SQL procedure successfully completed.

SQL>
SQL> BEGIN
  2      GENERATE_NEAREST_PRIME(9);
  3  END;
  4  /
7 is nearest prime less than n
11 is nearest prime greater than n
```

Task C

In this task, we were asked to create a procedure to take a string as input and show the string appending space after each character.

I created a procedure which takes a string as input. Then I appended each character of that string along with an extra space into a new empty string and printed that.

Then I took another empty string and appended the main string in reverse and make a reversed string. Then I match the main string and the reversed string to check If the given string is palindrome or not.

Below is the implementation of the procedure :

```
SET SERVEROUTPUT ON;

CREATE OR REPLACE PROCEDURE PRINT_AND_CHECK(
    S IN VARCHAR
) IS
    L   NUMBER;
    P   VARCHAR(20);
    R   VARCHAR2(20);
    PRI NUMBER;
BEGIN
    L := LENGTH(S);
    P := SUBSTR(S, 1, 1);
    FOR I IN 2..L LOOP
        P := P
            || RPAD(' ', 1, ' ')
               || SUBSTR(S, I, 1);
    END LOOP;
    DBMS_OUTPUT.PUT_LINE('Spaced string : ');
    DBMS_OUTPUT.PUT_LINE(P);
 -- IS PALLINDROME
    FOR I IN REVERSE 1.. L LOOP
        R := R
            || SUBSTR(S, I, 1);
    END LOOP;
    IF (S = R) THEN
        DBMS_OUTPUT.PUT_LINE('YES');
    ELSE
        DBMS_OUTPUT.PUT_LINE('NO');
    END IF;
END PRINT_AND_CHECK;
/
```

Result:

```
SQL> BEGIN
  2      PRINT_AND_CHECK('amina');
  3  END;
  4  /
Spaced string :
a m i n a
NO

PL/SQL procedure successfully completed.

SQL>
SQL> BEGIN
  2      PRINT_AND_CHECK('racecar');
  3  END;
  4  /
Spaced string :
r a c e c a r
YES

PL/SQL procedure successfully completed.

SQL>
SQL> BEGIN
  2      PRINT_AND_CHECK('forof');
  3  END;
  4  /
Spaced string :
f o r o f
YES
```