CSE 4508 – RDBMS Programming Lab

Lab 6

Prerequisites: Oracle 10g Express Edition, Any Text Editor (or SQL Developer)

Instructor: Shahriar Ivan

A. A hacker stumbles onto a database table containing only two columns: Username (varchar2) and Password_Length (number). Password_Length only contains a number, such as 7 or 8, denoting how long the password of that username is. Write a block of PL/SQL, using a function if necessary, which will first find the highest Password_Length from the table. It will then find out how many permutations the hacker needs to go through to crack that password. (The password only contains letters of the alphabet. You therefore have 26x2= 52 possibilities for each symbol of the password. However, no character can be repeated. So the correct answer, for a password of length 4, is: 52x51x50x49 = 6497400)

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. For instance, if the input string is “racecar”, the output will be “r a c e c a r”

2) Check if the original input string was a palindrome. Print “Yes” or “No” accordingly. [For example: ‘racecar’ is a palindrome, but ‘tracecar’ is not]

Task 1

```
set serveroutput on;
-- Task 1
create table Users(
    username varchar2(20),
    password_length number,
    constraint pk_username primary key(username)
);
insert into Users values('ash',10);
insert into Users values('asha',11);
insert into Users values('ashar',7);
insert into Users values('asharo',4);


create or replace function highestPassLength
return number is
    high number := 0;
begin
    select max(password_length) into high from Users;
    return high;
end;
/
create or replace function findPermutation(pass_len in number)
return number is
    fact number :=1;
    num number :=52;
```

```
begin
    for count in 1..pass_len
    loop
        fact:=num * fact;
        num:=num -1;
    end loop;
    return fact;
end;
/

-- main
declare
    len number;
    perm number;
begin
    len:= highestPassLength();
    perm:= findPermutation(len);

    dbms_output.put_line('Highest Password Length: ' || len);
    dbms_output.put_line('Total Permutations : ' || perm);
end;
/
```

Task 2

```
create or replace procedure PrintSpace(str in varchar2)
is
    ret varchar2(40);
begin
    ret := '';
    for i in 1..length(str)
    loop
        ret:=ret || ' ' || substr(str,i,1);
    end loop;
    dbms_output.put_line('Spaced Output : ' || ret);
end;
/


create or replace procedure PrintPalindrome(str in varchar2)
is
    rev varchar2(40);
begin
    rev := substr(str,length(str),1);
    for i in reverse 1..length(str)-1
    loop
        rev:= rev || substr(str,i,1);
    end loop;
    if rev = str then
        dbms_output.put_line('yes');
    else
```

```
        dbms_output.put_line('no');
    end if;
end;
/


--main
declare
    str varchar2(20) := '&string';
begin
    PrintSpace(str);
    PrintPalindrome(str);
end;
/
```