CSE 4508

Lab Report:1
Student ID: 210041264
Section: 2B

Date of Submission: September 22, 2024

1 Task: 1

Task 1 is about how many permutation a hacker needs to crack password. A table contains username and his/her password length information. I made a function which takes maximum password length value and returns total attempts needed. For loop has been used inside this function. Code:

```
set serveroutput on;

--1
create or replace function permutation (n INTEGER) return INTEGER
is
alpha integer := 52;
total integer := 1;
begin
  for i in 1..n loop
     total := total*alpha;
     alpha:=alpha-1;
 end loop;
 return total;
end;
/

declare
total integer;
begin
  select max(password_length) into total from plinfo where rownum=1;
  DBMS_output.put_line('Hacker needs maximum ' || permutation(total) || ' attempts');
end;
/
```

Output:

```
SQL> --1
SQL> create or replace function permutation (n INTEGER) return INTEGER
  2  is
  3  alpha integer := 52;
  4  total integer := 1;
  5  begin
  6    for i in 1..n loop
  7       total := total*alpha;
  8       alpha:=alpha-1;
  9   end loop;
 10   return total;
 11  end;
 12  /

Function created.

SQL>
SQL> declare
  2  total integer;
  3  begin
  4    select max(password_length) into total from plinfo where rownum=1;
  5    DBMS_output.put_line('Hacker needs maximum ' || permutation(total) || ' attempts');
  6  end;
  7  /

PL/SQL procedure successfully completed.

SQL> set serveroutput on;
SQL> declare
  2  total integer;
  3  begin
  4    select max(password_length) into total from plinfo where rownum=1;
  5    DBMS_output.put_line('Hacker needs maximum ' || permutation(total) || ' attempts');
  6  end;
  7  /
Hacker needs maximum 30342338208000 attempts
```

2 Task: 2

Task 2 is about inserting space between 2 characters of a string and checking if the string is palindrome or not. I created a procedure which runs a loop to adds character of the given string alongside a space in a new string and prints it. Also it has another loop to reverse the given string and checks with original string, if matches then it prints yes, else no. Code:

```
create or replace procedure add_space_is_palindrome(str in varchar2) is
s_str varchar2(1000) := '';
r_str varchar2(1000) := '';
begin
   for i in 1..length(str) loop
     s_str:= s_str || substr(str,i,1);
     if i<length(str) then
       s_str := s_str || ' ';
     end if;
   end loop;
   dbms_output.put_line(s_str);
   for i in Reverse 1..length(str) loop
     r_str := r_str || substr(str,i,1);
     end loop;
   if str=r_str then
     dbms_output.put_line('YES');
   else
     dbms_output.put_line('NO');
  end if;
end;
/

BEGIN
add_space_is_palindrome('racecar');
END;
/
```

Output:

```
SQL> create or replace procedure add_space_is_palindrome(str in varchar2) is
  2  s_str varchar2(1000) := '';
  3  r_str varchar2(1000) := '';
  4  begin
  5     for i in 1..length(str) loop
  6       s_str:= s_str || substr(str,i,1);
  7       if i<length(str) then
  8         s_str := s_str || ' ';
  9       end if;
 10     end loop;
 11     dbms_output.put_line(s_str);
 12     for i in Reverse 1..length(str) loop
 13       r_str := r_str || substr(str,i,1);
 14       end loop;
 15     if str=r_str then
 16       dbms_output.put_line('YES');
 17     else
 18       dbms_output.put_line('NO');
 19    end if;
 20  end;
 21  /

Procedure created.

SQL>
SQL> BEGIN
  2  add_space_is_palindrome('racecar');
  3  END;
  4  /
r a c e c a r
YES
```

3 Task: 3.1

Task 3.1 is about creating table and I did with simple table creation query. Code:

```
CREATE TABLE Library_Borrowing(
 B_ID Number not null,
 Name Varchar2(30) not null,
 Book_title VArchar2(30) not null,
 Borrow_Date Date not null,
 Due_Date Date Not null,
 Returned_Date Date
);
```

Output:

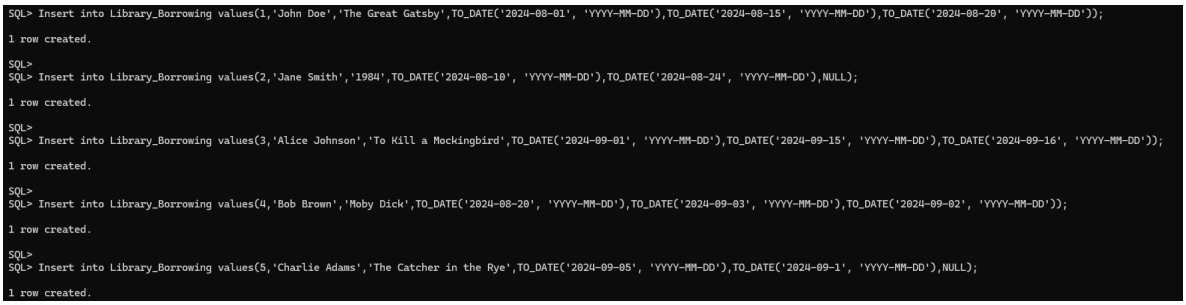
```
SQL> CREATE TABLE Library_Borrowing(
  2     B_ID Number not null,
  3     Name Varchar2(30) not null,
  4     Book_title VArchar2(30) not null,
  5     Borrow_Date Date not null,
  6     Due_Date Date Not null,
  7     Returned_Date Date
  8  );

Table created.
```

4 Task: 3.2

It is about inserting some values and I did it using simple sql queries. Code:

```
Insert into Library_Borrowing values(1,'John Doe','The Great Gatsby',TO_DATE('2024-08-01', 'YYYY-MM-D

Insert into Library_Borrowing values(2,'Jane Smith','1984',TO_DATE('2024-08-10', 'YYYY-MM-DD'),TO_DAT

Insert into Library_Borrowing values(3,'Alice Johnson','To Kill a Mockingbird',TO_DATE('2024-09-01',

Insert into Library_Borrowing values(4,'Bob Brown','Moby Dick',TO_DATE('2024-08-20', 'YYYY-MM-DD'),TO

Insert into Library_Borrowing values(5,'Charlie Adams','The Catcher in the Rye',TO_DATE('2024-09-05',
```

Output:



5 Task: 3.3

This task is about calculating Late fee. I wrote a function which takes due date and return date, returns time gap between them, if the book is not returned, it subtracts due date from current date (sysdate). Code:

```
create or replace function Calculate_Late_Fee (dd DATE,rd DATE) return INTEGER
is
late_fee integer := 0;
begin
  if rd is NULL and dd is NULL then
    return NULL;
  end if;
  if rd is NULL then
    late_fee := TRUNC(sysdate-dd);
  else
    late_fee := TRUNC(rd-dd);
  end if;
 if late_fee<0 then
   return 0;
 else
   return late_fee;
 end if;
end;
/

declare
dd DATE;
rd DATE;
book_name varchar2(30) :='ZARIF';
```

```
begin
  select Due_Date,Returned_Date into dd, rd from Library_Borrowing where Book_title=book_name;
  DBMS_output.put_Line('Late Fee: ' || Calculate_Late_Fee(dd,rd) || ' $');
EXCEPTION
  When NO_DATA_FOUND then
  DBMS_output.put_Line('No record found for the book: ' || book_name);
end;
/
```

Output:

```
SQL> create or replace function Calculate_Late_Fee (dd DATE,rd DATE) return INTEGER
  2  is
  3  late_fee integer := 0;
  4  begin
  5    if rd is NULL and dd is NULL then
  6      return NULL;
  7    end if;
  8    if rd is NULL then
  9      late_fee := TRUNC(sysdate-dd);
 10    else
 11      late_fee := TRUNC(rd-dd);
 12    end if;
 13   if late_fee<0 then
 14     return 0;
 15   else
 16     return late_fee;
 17   end if;
 18  end;
 19  /

Function created.

SQL>
SQL> declare
  2  dd DATE;
  3  rd DATE;
  4  book_name varchar2(30) :='The Catcher in the Rye';
  5  begin
  6    select Due_Date,Returned_Date into dd, rd from Library_Borrowing where Book_title=book_name;
  7    DBMS_output.put_Line('Late Fee: ' || Calculate_Late_Fee(dd,rd) || ' $');
  8  EXCEPTION
  9    When NO_DATA_FOUND then
 10    DBMS_output.put_Line('No record found for the book: ' || book_name);
 11  end;
 12  /
Late Fee: 21 $

PL/SQL procedure successfully completed.
```

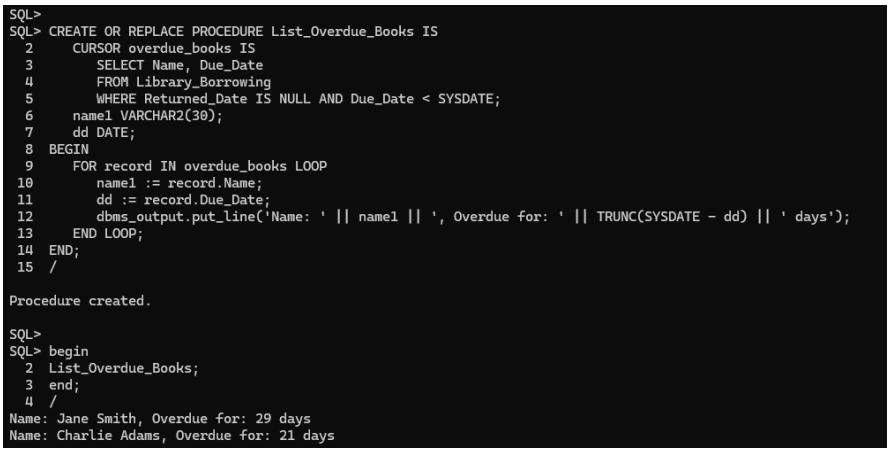
6 Task: 3.4

This task is about counting overdue date numbers. I created a procedure which iterates through every rows and if any book is not returned till now, it counts how many days have past since due date (sysdate-duedate). If any due date has not yet arrived, it is not overdue, then the procedure prints name and the how many days the book is overdue. Code:

```
CREATE OR REPLACE PROCEDURE List_Overdue_Books IS
   CURSOR overdue_books IS
      SELECT Name, Due_Date
      FROM Library_Borrowing
      WHERE Returned_Date IS NULL AND Due_Date < SYSDATE;
   name1 VARCHAR2(30);
   dd DATE;
BEGIN
   FOR record IN overdue_books LOOP
      name1 := record.Name;
      dd := record.Due_Date;
      dbms_output.put_line('Name: ' || name1 || ', Overdue for: ' || TRUNC(SYSDATE - dd) || ' days');
   END LOOP;
END;
/
```

```
begin
List_Overdue_Books;
end;
/
```

Output:



```
SQL>
SQL> CREATE OR REPLACE PROCEDURE List_Overdue_Books IS
  2     CURSOR overdue_books IS
  3        SELECT Name, Due_Date
  4        FROM Library_Borrowing
  5        WHERE Returned_Date IS NULL AND Due_Date < SYSDATE;
  6     name1 VARCHAR2(30);
  7     dd DATE;
  8  BEGIN
  9     FOR record IN overdue_books LOOP
 10        name1 := record.Name;
 11        dd := record.Due_Date;
 12        dbms_output.put_line('Name: ' || name1 || ', Overdue for: ' || TRUNC(SYSDATE - dd) || ' days');
 13     END LOOP;
 14  END;
 15  /

Procedure created.

SQL>
SQL> begin
  2  List_Overdue_Books;
  3  end;
  4  /
Name: Jane Smith, Overdue for: 29 days
Name: Charlie Adams, Overdue for: 21 days
```