Institute of Engineering & Management
Department of Computer Science & Engineering
Data-Base Management System Lab for 3rd year 6th semester 2019
Code: CS 691

Date: 28/03/19

WEEK-5

Problem Statement-1: display your name 5 times using for loop.

SQL :

```
SQL> set serveroutput on;
SQL> DECLARE
  2     name varchar2(20) := 'Ranajit Roy';
  3  BEGIN
  4     for i in 1..5 LOOP
  5             dbms_output.put_line(name);
  6     END LOOP;
  7  END;
  8  /
Ranajit Roy
Ranajit Roy
Ranajit Roy
Ranajit Roy
Ranajit Roy
```

Problem Statement-2: write a pl/sql block of code to invert a number 12345 to 54321.

SQL :

```
SQL> DECLARE
  2     num integer := &num;
  3     result integer := 0;
  4  BEGIN
  5     WHILE num!=0 LOOP
  6             result := result*10 + MOD(num,10);
  7             num := FLOOR(num/10);
  8     END LOOP;
  9     dbms_output.put_line(result);
 10  END;
 11  /
Enter value for num: 12345
old   2:        num integer := &num;
new   2:        num integer := 12345;
54321
```

Problem Statement-3: write a pl/sql code block to calculate the area of circle for a value of radius varying from 3 to 7. store the radius & the corresponding values of calculated area in an empty table named areas, consisting of two columns, radius & area.

SQL :

```
SQL> create table area(radius integer, area number(5,4));

Table created.

SQL> DECLARE
  2     pi constant number(5,2) := 3.14;
  3     area number(5,2);
  4  BEGIN
```

```
  5     FOR radius in 3..7 LOOP
  6             area := pi*radius*radius;
  7             insert into area values(radius, area);
  8     END LOOP;
  9  END;
 10  /

PL/SQL procedure successfully completed.

SQL> select * from area;

    RADIUS       AREA
---------- ----------
         3      28.26
         4      50.24
         5       78.5
         6     113.04
         7     153.86
```

Problem Statement-4: create a simple loop such that a message is displayed when a loop exceeds a particular value(while loop).

SQL :

```
SQL> DECLARE
  2     inp integer := &inp;
  3     a integer := 0;
  4  BEGIN
  5     WHILE a<=inp LOOP
  6             dbms_output.put_line(a);
  7             a := a+1;
  8     END LOOP;
  9     dbms_output.put_line('Value Exceeded!');
 10  END;
 11  /
Enter value for inp: 4
old   2:        inp integer := &inp;
new   2:        inp integer := 4;
0
1
2
3
4
Value Exceeded!
```

Problem Statement-5: write a pl/sql block code that will accept an account number from the user, check if the user's balance is less than the minimum balance, only then deduct rs. 100/- from the available balance. the process is fired on the accounts table.

SQL :

```
SQL> select * from accounts;

     ACCNO    CURRBAL
---------- ----------
         1       4900
         2      19900
         3      34900
         4      49900
         5      64900
         6      79900
```

```
6 rows selected.

SQL> DECLARE
  2     acid accounts.accno%type := &acid;
  3     bal accounts.currbal%type;
  4  BEGIN
  5     SELECT currbal INTO bal FROM accounts WHERE accno=acid;
  6     IF bal<=50000 THEN
  7             UPDATE accounts SET currbal = currbal-100 WHERE
accno=acid;
  8             dbms_output.put_line('Balance updated!');
  9     ELSE
 10             dbms_output.put_line('Balance above min balance.');
 11     END IF;
 12  END;
 13  /
Enter value for acid: 4
old   2:        acid accounts.accno%type := &acid;
new   2:        acid accounts.accno%type := 4;
Balance updated!

PL/SQL procedure successfully completed.

SQL> select * from accounts;

     ACCNO    CURRBAL
---------- ----------
         1       4900
         2      19900
         3      34900
         4      49800
         5      64900
         6      79900

6 rows selected.
```

Problem Statement-6: bank declares 8% interest on capital. so, update all accounts using pl/sql code block.

SQL :

```
SQL> DECLARE
  2  BEGIN
  3     UPDATE accounts SET currbal=currbal*1.08;
  4  END;
  5  /

PL/SQL procedure successfully completed.

SQL> select * from accounts;

     ACCNO    CURRBAL
---------- ----------
         1       5292
         2      21492
         3      37692
         4      53784
         5      70092
         6      86292

6 rows selected.
```