Rajalakshmi Engineering College

Rajalakshmi Nagar, Thandalam, Chennai - 602 105

Department of Computer Science and Engineering

CS6312 - Database Management Systems Lab

Sample Programs

PL / SQL

(Regulation - 2013)

Prepared by:
B.BHUVANESWARAN
Assistant Professor (SS) / CSE / REC
bhuvaneswaran@rajalakshmi.edu.in

Q-01

Write a pl/sql program to find the largest of two numbers.

```
SQL> ed largetwo

declare
     a number;
     b number;
begin
     a := &a;
     b := &b;
     if(a > b) then
          dbms_output.put_line('a is larger ' || a);
     else
          dbms_output.put_line('b is larger ' || b);
     end if;
end;
/

SQL> set serveroutput on

SQL> @largetwo
Enter value for a: 10
old   5:          a := &a;
new   5:          a := 10;
Enter value for b: 20
old   6:          b := &b;
new   6:          b := 20;
b is larger 20

PL/SQL procedure successfully completed.
```

Q-02

Write PL/SQL Program to generate even numbers.

```
SQL> ed evennos

declare
     n number;
begin
     n := &n;
     for i in 0..n
     loop
          dbms_output.put_line(i * 2);
     end loop;
end;
/

SQL> set serveroutput on

SQL> @evennos
Enter value for n: 5
old   5:         n := &n;
new   5:         n := 5;
0
2
4
6
8
10

PL/SQL procedure successfully completed.
```

Q-03

Write a pl/sql program to find the factorial of a given number using functions.

```
SQL> ed factfn

create or replace function fact(n number)
return number is
i number;
f number := 1;
begin
     for i in 1..n
     loop
          f := f * i;
     end loop;
     return(f);
end fact;
/

SQL> @factfn

Function created.

SQL> ed factorial

declare
     f number;
     n number := &n;
begin
     f := fact(n);
     dbms_output.put_line('Factorial = ' || f);
end;
/

SQL> set serveroutput on

SQL> @factorial
Enter value for n: 5
old   3:        n number := &n;
new   3:        n number := 5;
Factorial = 120

PL/SQL procedure successfully completed.
```

Q-04

Write a PL/SQL program to find the greatest of 3 numbers.

```
SQL> ed greatthree

declare
     a number;
     b number;
     c number;
begin
     a := &a;
     b := &b;
     c := &c;
     if(a > b) and (a > c) then
          dbms_output.put_line('A is greater ' || a);
     elsif(b > a) and (b > c) then
          dbms_output.put_line('B is greater ' || b);
     else
          dbms_output.put_line('C is greater ' || c);
     end if;
end;
/

SQL> set serveroutput on

SQL> @greatthree
Enter value for a: 20
old   6:        a := &a;
new   6:        a := 20;
Enter value for b: 30
old   7:        b := &b;
new   7:        b := 30;
Enter value for c: 10
old   8:        c := &c;
new   8:        c := 10;
B is greater 30

PL/SQL procedure successfully completed.
```

Q-05

Write a pl/sql program to find the summation of odd numbers using for loop.

```
SQL> ed oddsum

declare
     n number;
     oddsum number := 0;
begin
     n := &n;
     for i in 1..n
     loop
          if mod(i, 2) = 1 then
               oddsum := oddsum + i;
          end if;
     end loop;
     dbms_output.put_line('Sum = ' || oddsum);
end;
/

SQL> set serveroutput on

SQL> @oddsum
Enter value for n: 10
old   5:        n := &n;
new   5:        n := 10;
Sum = 25

PL/SQL procedure successfully completed.
```

Q-06

Write a pl/sql program to swap two numbers.

```
SQL> ed swap

declare
     a number;
     b number;
     c number;
begin
     a := &a;
     b := &b;
     dbms_output.put_line('Before Swap : ');
     dbms_output.put_line('a = ' || a || ' and b = ' || b);
     c := a;
     a := b;
     b := c;
     dbms_output.put_line('After Swap : ');
     dbms_output.put_line('a = ' || a || ' and b = ' || b);
end;
/

SQL> set serveroutput on

SQL> @swap
Enter value for a: 10
old   6:          a := &a;
new   6:          a := 10;
Enter value for b: 20
old   7:          b := &b;
new   7:          b := 20;
Before Swap :
a = 10 and b = 20
After Swap :
a = 20 and b = 10

PL/SQL procedure successfully completed.
```

Q-07

Write a pl/sql program to check whether the given number is prime or not.

```
SQL> ed prime

declare
     n number;
     p number := 1;
begin
     n := &n;
     for i in 2..n/2
     loop
          if mod(n, i) = 0 then
               p:= 0;
          end if;
     end loop;

     if p = 1  then
          dbms_output.put_line(n || ' is a Prime Number');
     else
          dbms_output.put_line(n || ' is not a Prime Number');
     end if;
end;
/

SQL> set serveroutput on

SQL> @prime
Enter value for n: 2
old   5:          n := &n;
new   5:          n := 2;
2 is a Prime Number

PL/SQL procedure successfully completed.

SQL> @prime
Enter value for n: 6
old   5:          n := &n;
new   5:          n := 6;
6 is not a Prime Number

PL/SQL procedure successfully completed.
```

Q-08

Write a pl/sql program to find the factorial of a given number.

```
SQL> ed fact

declare
	n number;
	f number := 1;
begin
	n := &n;
	for i in 1..n
	loop
		f := f * i;
	end loop;
	dbms_output.put_line('Factorial of ' || n || ' is ' || f);
end;
/

SQL> set serveroutput on

SQL> @fact
Enter value for n: 5
old   5:          n := &n;
new   5:          n := 5;
Factorial of 5 is 120

PL/SQL procedure successfully completed.
```

Q-09

Write a pl/sql program to find the sum of 1-100 numbers.

```
SQL> ed sumnos

declare
     sumnos number := 0;
begin
     for i in 1..100
     loop
          sumnos := sumnos + i;
     end loop;
     dbms_output.put_line('Sum of nos. 1-100 is ' || sumnos);
end;
/

SQL> set serveroutput on

SQL> @sumnos
Sum of nos. 1-100 is 5050

PL/SQL procedure successfully completed.
```

Q-10

write a pl/sql code to accept the text and check the text is palindrome or not.

```
SQL> ed palindrome

declare
     len number;
     str1 varchar(20);
     str2 varchar(20);
begin
     str1 := '&str1';
     len := length(str1);
     for i in reverse 1..len
     loop
          str2 := str2 || substr(str1, i, 1);
     end loop;
     if str2 = str1 then
             dbms_output.put_line(str1 || 'is a Palindrome');
     else
          dbms_output.put_line(str1 || 'is not a Palindrome');
     end if;
end;
/

SQL> set serveroutput on

SQL> @palindrome
Enter value for str1: chennai
old   6:        str1 := '&str1';
new   6:        str1 := 'chennai';
chennaiis not a Palindrome

PL/SQL procedure successfully completed.

SQL> @palindrome
Enter value for str1: madam
old   6:        str1 := '&str1';
new   6:        str1 := 'madam';
madamis a Palindrome

PL/SQL procedure successfully completed.
```

Q-11

Write a pl/sql program to generate numbers from 0 to 25 in step of 5.

```
SQL> ed gennos

declare
     i number := 0;
begin
     while(i <= 25)
     loop
             dbms_output.put_line(i);
          i := i + 5;
     end loop;
end;
/

SQL> set serveroutput on

SQL> @gennos
0
5
10
15
20
25

PL/SQL procedure successfully completed.
```

Q-12

Write a PL/SQL to find and display the sum of first n natural numbers.

```
SQL> ed naturalnos

declare
     n number;
     sumnat number := 0;
begin
     n := &n;
     for i in 1..n
     loop
          sumnat := sumnat + i;
     end loop;
     dbms_output.put_line('Sum of first ' || n || ' no. is ' || sumnat);
end;
/

SQL> set serveroutput on

SQL> @naturalnos
Enter value for n: 5
old   5:        n := &n;
new   5:        n := 5;
Sum of first 5 no. is 15

PL/SQL procedure successfully completed.
```

Q-13

Write a PL/SQL to generate Fibonacci series.

```
SQL> ed fibonacci

declare
     a number := -1;
     b number := 1;
     c number := 0;
     n number;
begin
     n := &n;
     for i in 1..n
     loop
          c := a + b;
          dbms_output.put_line(c);
          a := b;
          b := c;
     end loop;
end;
/

SQL> set serveroutput on

SQL> @fibonacci
Enter value for n: 5
old   7:          n := &n;
new   7:          n := 5;
0
1
1
2
3

PL/SQL procedure successfully completed.
```

Q-14

Write a PL/SQL to check the given number is Armstrong number or not.

```
SQL> ed armstrong

declare
     n number;
     r number;
     s number := 0;
     t number;
begin
     n := &n;
     t := n ;
     while (n > 0)
     loop
          r := mod(n, 10);
          s := s + (r * r *  r) ;
          n := floor(n / 10) ;
     end loop;
     if s = t then
          dbms_output.put_line(t || ' is an armstrong number') ;
     else
          dbms_output.put_line(t || ' is not an armstrong number') ;
     end if ;
end;
/

SQL> set serveroutput on

SQL> @armstrong
Enter value for n: 143
old   7:          n := &n;
new   7:          n := 143;
143 is not an armstrong number

PL/SQL procedure successfully completed.

SQL> @armstrong
Enter value for n: 153
old   7:          n := &n;
new   7:          n := 153;
153 is an armstrong number

PL/SQL procedure successfully completed.
```

RAJALAKSHMI
ENGINEERING COLLEGE

Website : www.rajalakshmi.org

Elearning : www.rajalakshmiengg.com

Give a man a fish and you feed him for a day.
Teach him how to fish and you feed him for a lifetime.