DBMS Practical 2

Dharmit Shah SYIT 330

Practical 2: Control Structures

1. Write a PL/SQL program to display the sum of first 10 integers.

```
SQL> set serveroutput  on
SQL> declare
  2  i int;
  3  s int;
  4  begin
  5  s:=0;
  6  for i in 1..10 loop
  7  s:=s+i;
  8  end loop;
  9  dbms_output.put_line('Sum of integer is = '||s);
 10  end;
 11  /
Sum of integer is = 55

PL/SQL procedure successfully completed.
```

2. Write a PL/SQL program to display the sum of first 10 odd numbers.

```
SQL> declare
  2  num NUMBER(3) := 1;
  3  sum1 NUMBER(4) := 0;
  4  begin
  5  while num <= 20 loop
  6  dbms_output.put_line(num);
  7  sum1 := sum1 + num;
  8  num := num + 2;
  9  end loop;
 10  dbms_output.put_line('Sum of first 10 odd numbers is '||sum1);
 11  end;
 12  /
1
3
5
7
9
11
13
15
17
19
Sum of first 10 odd numbers is 100

PL/SQL procedure successfully completed.
```

3. Write a PL/SQL program to calculate the area of a circle and insert the area and radius in a table aoc (sno,radius,area) till radius is less than 10.

Run SQL Command Line

```
SQL> create table aoc_dharmit(sno int,radius int, area float);

Table created.

SQL> declare
  2  sno int := 1;
  3  radius int := 1;
  4  area float;
  5  begin
  6  while radius < 10
  7  loop
  8  area := 3.14*radius*radius;
  9  insert into aoc_dharmit values(sno, radius, area);
 10  sno := sno+1;
 11  radius := radius + 1;
 12  end loop;
 13  end;
 14  /

PL/SQL procedure successfully completed.

SQL> select * from aoc_dharmit;

       SNO     RADIUS       AREA
---------- ---------- ----------
         1          1       3.14
         2          2      12.56
         3          3      28.26
         4          4      50.24
         5          5       78.5
         6          6     113.04
         7          7     153.86
         8          8     200.96
         9          9     254.34

9 rows selected.

SQL>
```

4. Write a PL/SQL program to reverse the number (234 as 432).

```
SQL> declare
  2  num NUMBER;
  3  rev NUMBER;
  4  begin
  5  num:=234;
  6  rev :=0;
  7  while num>0 loop
  8  rev:=(rev*10) + mod(num,10);
  9  num:= floor(num/10);
 10  end loop;
 11  dbms_output.put_line('Reverse of the number is: '||rev);
 12  end;
 13  /
Reverse of the number is: 432

PL/SQL procedure successfully completed.
```

5. Write a PL/SQL program to print the length of entered string.

```
SQL> declare
  2  st varchar(30):= '&st';
  3  begin
  4  dbms_output.put_line('Length of the string is: '||(Length(st)));
  5  end;
  6  /
Enter value for st: dharmit shah
old   2: st varchar(30):= '&st';
new   2: st varchar(30):= 'dharmit shah';
Length of the string is: 12

PL/SQL procedure successfully completed.
```

6. Write a PL/SQL program to count number of employees in dept 10, and if they are greater than 3 print the count otherwise do nothing.

```
declare
e int;
begin
select count(*) into e from emp_dharmit where deptno = 10;
if (e > 3) then
dbms_output.put_line('Total number of employee in deptno 10 is '|| e);
end if;
end;
/
```

Results Explain Describe Saved SQL History

```
Statement processed.
```

0.58 seconds

7. Write a PL/SQL block to find the factorial of a number.

```
SQL> declare
  2  n number:=&n;
  3  fac number:=1;
  4  i number;
  5  begin
  6  n:=n;
  7  for i in 1..n
  8  loop
  9  fac:=fac*i;
 10  end loop;
 11  dbms_output.put_line('Factorial is '||fac);
 12  end;
 13  /
Enter value for n: 6
old   2: n number:=&n;
new   2: n number:=6;
Factorial is 720

PL/SQL procedure successfully completed.
```

8. Write a block to display the number from 1 to 10 using unconstraint loop.

```
SQL> declare
  2  num int:=1;
  3  begin
  4  loop
  5  dbms_output.put_line(num);
  6  num := num +1;
  7  exit when num=11;
  8  end loop;
  9  end;
 10  /
1
2
3
4
5
6
7
8
9
10

PL/SQL procedure successfully completed.
```

9. Write a PL/SQL block using CASE statement to accept the owner name from the user. The user name can be SYS, SYSTEM, HR or SCOTT. If the owner name is SYS then print the result is 'The Owner is SYS'. If the owner name is SYSTEM then print the result is 'The Owner is SYSTEM'. If the owner name is HR then print the result is 'The Owner is HR'. If the owner name is SCOTT then print the result is 'The Owner is SCOTT'. Otherwise print 'Invalid Choice'.

```
SQL> declare
  2  o char(20) := '&ownername';
  3  begin
  4  case o
  5  when 'SYS' then dbms_output.put_line('The Owner is SYS');
  6  when 'SYSTEM' then dbms_output.put_line('The Owner is SYSTEM');
  7  when 'HR' then dbms_output.put_line('The Owner is HR');
  8  when 'SCOTT' then dbms_output.put_line('The Owner is SCOTT');
  9  else dbms_output.put_line('Invalid Choice');
 10  end case;
 11  end;
 12  /
Enter value for ownername: dharmit
old   2: o char(20) := '&ownername';
new   2: o char(20) := 'dharmit';
Invalid Choice

PL/SQL procedure successfully completed.
```

10. Write a PL/SQL block to find factorial of a number which is accepted by the user and store it under the table fac(num,fact).

```
SQL> create table fac(num int, fact int);

Table created.

SQL> declare
  2  fact int :=1;
  3  num int := &num;
  4  val int;
  5  begin
  6  val := num;
  7  while val>0 loop
  8  fact := val * fact;
  9  val := val-1;
 10  end loop;
 11  insert into fac values(num,fact);
 12  end;
 13  /
Enter value for num: 8
old   3: num int := &num;
new   3: num int := 8;

PL/SQL procedure successfully completed.
```

```
 13  /
Enter value for num: 6
old   3: num int := &num;
new   3: num int := 6;

PL/SQL procedure successfully completed.

SQL> select * from fac;

       NUM       FACT
---------- ----------
         8      40320
         8      40320
         6        720
```

11. Write a PL/SQL to read a number and check whether it is greater than 100 or not and print appropriate message.

```
SQL> declare
  2  num int := &num;
  3  begin
  4  if (num > 100) then
  5  dbms_output.put_line('The number is greater then 100');
  6  else
  7  dbms_output.put_line('The number is less then 100');
  8  end if;
  9  end;
 10  /
Enter value for num: 330
old   2: num int := &num;
new   2: num int := 330;
The number is greater then 100

PL/SQL procedure successfully completed.
```

12. Write a PL/SQL to read the salary of an employee 7900 and display the appropriate message if it lies in the range of 1000 and 5000.

```
declare
salary int;
begin
select sal into salary from emp_dharmit where empno=7900;
if(salary > 1000 and salary < 5000) then
dbms_output.put_line('The salary of the employee 7900 is '|| salary);
end if;
end;
/
```

Results Explain Describe Saved SQL History

```
Statement processed.
```

13. Write a PL/SQL to swap two numbers and display the swapped numbers.

```
SQL> declare
  2  num1 int;
  3  num2 int;
  4  temp int;
  5  begin
  6  num1 := 100;
  7  num2 := 200;
  8  dbms_output.put_line('Before swap');
  9  dbms_output.put_line('num1 = '|| num1 ||'num2 = '||num2);
 10  temp := num1;
 11  num1 := num2;
 12  num2 := temp;
 13  dbms_output.put_line('After swap');
 14  dbms_output.put_line('num1 = '|| num1 ||'num2 = '||num2);
 15  end;
 16  /
Before swap
num1 = 100num2 = 200
After swap
num1 = 200num2 = 100

PL/SQL procedure successfully completed.
```

14. Write a PL/SQL block to update the salary of the employee with 1000 when total number of employees in a particular department is greater than 3.

```
declare
s int;
dpno int:=:dpno;
begin
select count(*) into s from emp_dharmit where deptno=dpno;
if(s>3) then
update emp_dharmit
set sal=sal + 1000
where deptno=dpno;
end if;
end;
/
```

Results Explain Describe Saved SQL History

Statement processed.

0.13 seconds

```
select sal from emp_dharmit where deptno=10;
```

Results Explain Describe Saved SQL History

SAL
5000
1300
2450

3 rows returned in 0.20 seconds Download

15. Write a PL/SQL block to delete the records of the table employee by accepting the table name from the user.

```
SQL> begin
  2  delete from &tablename;
  3  end;
  4  /
Enter value for tablename: dharmit
old   2: delete from &tablename;
new   2: delete from dharmit;

PL/SQL procedure successfully completed.
```

16. Write a PL/SQL to check whether the character entered is a vowel or not.

```
SQL> declare
  2  ch char(3) := '&character';
  3  begin
  4  case ch
  5  when 'a' then dbms_output.put_line('vowel');
  6  when 'A' then dbms_output.put_line('vowel');
  7  when 'e' then dbms_output.put_line('vowel');
  8  when 'E' then dbms_output.put_line('vowel');
  9  when 'i' then dbms_output.put_line('vowel');
 10  when 'I' then dbms_output.put_line('vowel');
 11  when 'o' then dbms_output.put_line('vowel');
 12  when 'O' then dbms_output.put_line('vowel');
 13  when 'u' then dbms_output.put_line('vowel');
 14  when 'U' then dbms_output.put_line('vowel');
 15  else dbms_output.put_line(ch ||' is not a vowel');
 16  end case;
 17  end;
 18  /
Enter value for character: d
old   2: ch char(3) := '&character';
new   2: ch char(3) := 'd';
d   is not a vowel

PL/SQL procedure successfully completed.
```

17. Write a PL/SQL to check whether a number is even or odd.

```
SQL> declare
  2  numb int := &numb;
  3  begin
  4  if(mod(numb, 2) = 0) then
  5  dbms_output.put_line(numb ||' is even number');
  6  else
  7  dbms_output.put_line(numb ||' is odd number');
  8  end if;
  9  end;
 10  /
Enter value for numb: 330
old   2: numb int := &numb;
new   2: numb int := 330;
330 is even number

PL/SQL procedure successfully completed.
```

18. Write a PL/SQL block using case statement to print the salary as high if it is greater then 10000, moderate if it is between 5000 and 10000 and low if it is less than 5000. The salary has been taken as an input of a specific employee whose empid is accepted by the user.

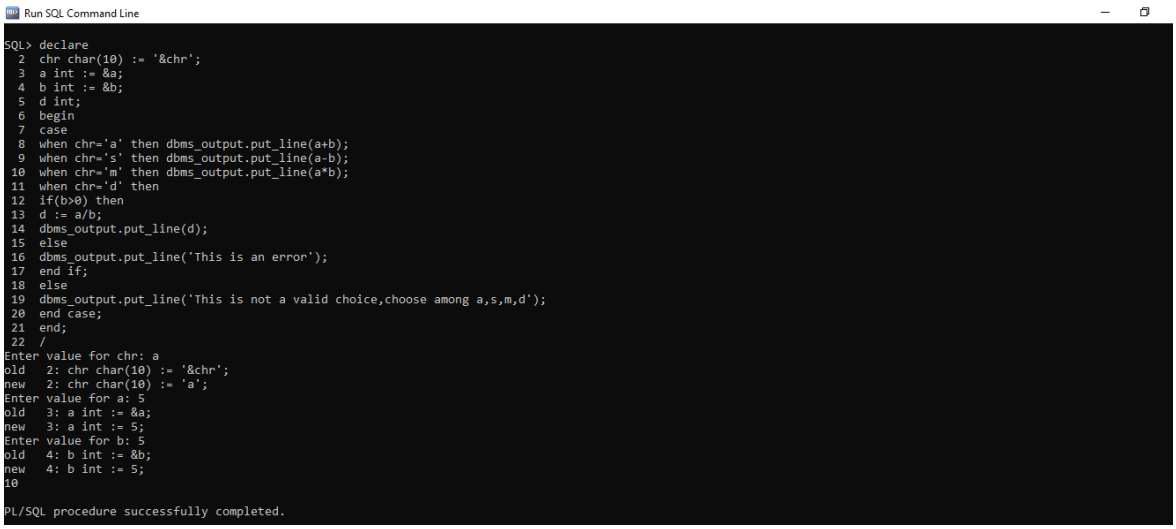
```
declare
s int;
eno int := 7934;
begin
select sal into s from emp_dharmit where empno=eno;
case
when s>10000 then dbms_output.put_line('Salary is higher');
when s>5000 and s<10000 then dbms_output.put_line('Salary is moderate');
else dbms_output.put_line('Salary is lower');
end case;
end;
/
```

Results Explain Describe Saved SQL History

```
Salary is lower

Statement processed.
```

19. Write a PL/SQL block using case statement to perform addition, subtraction, multiplication and division for the individual choices a ,s ,m, d. The division can only take place if the divisor is greater than 0 else error message should be printed.



20. Write a PL/SQL block to print the numbers from 1 to 10 using While and For Loop.

```
SQL> declare
  2  i int := 1;
  3  begin
  4  while i <=10 loop
  5  dbms_output.put_line(i);
  6  i := i+1;
  7  end loop;
  8  end;
  9  /
1
2
3
4
5
6
7
8
9
10

PL/SQL procedure successfully completed.

SQL> declare
  2  i int;
  3  begin
  4  for i in 1..10 loop
  5  dbms_output.put_line(i);
  6  end loop;
  7  end;
  8  /
1
2
3
4
5
6
7
8
9
10

PL/SQL procedure successfully completed.
```

21. Write a PL/SQL block to print the Fibonacci series up to 10.

```
SQL> declare
  2  first number := 0;
  3  second number := 1;
  4  temp number;
  5  n number := &n;
  6  i number;
  7  begin
  8  dbms_output.put_line('Series:');
  9  dbms_output.put_line(first);
 10  dbms_output.put_line(second);
 11  for i in 2..n
 12  loop
 13  temp:=first+second;
 14  first := second;
 15  second := temp;
 16  dbms_output.put_line(temp);
 17  end loop;
 18  end;
 19  /
Enter value for n: 10
old   5: n number := &n;
new   5: n number := 10;
Series:
0
1
1
2
3
5
8
13
21
34
55

PL/SQL procedure successfully completed.
```

22. Write a PL/SQL block to calculate the area of a circle till radius less than 10.

```
SQL> declare
  2  sno int:= 1;
  3  radius int := 1;
  4  area float;
  5  begin
  6  while radius<10
  7  loop
  8  area := 3.14*radius*radius;
  9  dbms_output.put_line(area);
 10  sno := sno + 1;
 11  radius := radius + 1;
 12  end loop;
 13  end;
 14  /
3.14
12.56
28.26
50.24
78.5
113.04
153.86
200.96
254.34

PL/SQL procedure successfully completed.
```

23. Write a PL/SQL block to display the number of employees when the deptno is inputted by the user.

```
declare
dno int:= 20;
employeeno int;
begin
select count(*) into employeeno from emp_dharmit where deptno=dno;
dbms_output.put_line(employeeno);
end;
/
```

Results Explain Describe Saved SQL History

```
5

Statement processed.
```

24. Write a PL/SQL block to print greatest among three numbers.

```
SQL> declare
  2  num1 int := &num1;
  3  num2 int := &num2;
  4  num3 int := &num3;
  5  begin
  6  if(num1>num2 and num1>num3) then
  7  dbms_output.put_line(num1 ||' is greatest from '||num2||' and '||num3);
  8  elsif(num2>num1 and num2>num3) then
  9  dbms_output.put_line(num2 ||' is greatest from '||num1||' and '||num3);
 10  else
 11  dbms_output.put_line(num3 ||' is greatest from '||num1||' and '||num2);
 12  end if;
 13  end;
 14  /
Enter value for num1: 10
old   2: num1 int := &num1;
new   2: num1 int := 10;
Enter value for num2: 20
old   3: num2 int := &num2;
new   3: num2 int := 20;
Enter value for num3: 30
old   4: num3 int := &num3;
new   4: num3 int := 30;
30 is greatest from 10 and 20

PL/SQL procedure successfully completed.
```

25. Write a PL/SQL block to display the appropriate day of the week according to the choice made by the user.

```
SQL> declare
  2  daynum int := &daynum;
  3  begin
  4  case daynum
  5  when 1 then dbms_output.put_line('Monday');
  6  when 2 then dbms_output.put_line('Tuesday');
  7  when 3 then dbms_output.put_line('Wednesday');
  8  when 4 then dbms_output.put_line('Thursday');
  9  when 5 then dbms_output.put_line('Friday');
 10  when 6 then dbms_output.put_line('Saturday');
 11  when 7 then dbms_output.put_line('Sunday');
 12  end case;
 13  end;
 14  /
Enter value for daynum: 4
old   2: daynum int := &daynum;
new   2: daynum int := 4;
Thursday

PL/SQL procedure successfully completed.
```

26. Create a PL/SQL block that has four sections. Each section should output a statement. Use labels and the Goto command to output the section messages in the following order:

Section 3

Section 2

Section 1

Section 4

```
SQL> begin
  2  dbms_output.put_line('section 3');
  3  goto section2;
  4  <<section1>>
  5  dbms_output.put_line('section 1');
  6  goto section4;
  7  <<section2>>
  8  dbms_output.put_line('section 2');
  9  goto section1;
 10  <<section4>>
 11  dbms_output.put_line('section 4');
 12  end;
 13  /
section 3
section 2
section 1
section 4

PL/SQL procedure successfully completed.
```

27. Write a PL/SQL block to check whether the entered year is a leap year or not.

```
SQL> declare
  2  year int := &year;
  3  begin
  4  if mod(year, 4)=0 and mod(year,100)!=0 or mod(year, 400)=0 then
  5  dbms_output.put_line(year ||' is a leap year');
  6  else
  7  dbms_output.put_line(year ||' is not a leap year');
  8  end if;
  9  end;
 10  /
Enter value for year: 2002
old   2: year int := &year;
new   2: year int := 2002;
2002 is not a leap year

PL/SQL procedure successfully completed.

SQL> /
Enter value for year: 2004
old   2: year int := &year;
new   2: year int := 2004;
2004 is a leap year

PL/SQL procedure successfully completed.
```

28. Write a PL/SQL block to display the numbers from 1 to 10 using EXIT and EXIT WHEN statement.

```
SQL> declare
  2  i int := 1;
  3  begin
  4  loop
  5  dbms_output.put_line(i);
  6  i := i + 1;
  7  exit when i = 11;
  8  end loop;
  9  end;
 10  /
1
2
3
4
5
6
7
8
9
10

PL/SQL procedure successfully completed.
```

29. Write a PL/SQL block to accept job from EMP table.

Give the following raise in the salary: -

By 9% if job is clerk.

By 8% if job is manager.

By 7% if job is salesman.

Update the salary of the EMP table.

```
declare
j varchar(30):= '&j';
begin
if(j='CLERK')then
update emp_dharmit
set sal = sal + sal*9/100 where job=j;
elsif(j='MANAGER') then
update emp_dharmit
set sal = sal + sal*8/100 where job=j;
elsif(j='SALESMAN') then
update emp_dharmit
set sal = sal + sal*7/100 where
job=j;
end if;
end;
/
```

Results Explain Describe Saved SQL History

```
Statement processed.
```

```
select sal,job from emp_dharmit where job='CLERK';
```

Results Explain Describe Saved SQL History

SAL	JOB
800	CLERK
1100	CLERK
950	CLERK
1300	CLERK

4 rows returned in 0.05 seconds Download

30. Write a PL/SQL block to get the details of marks (rollno, marks1, marks2, grade) out of 100 for marks1 and marks2 respectively. Display the grade in table marks using if statement as specified below

If stud_pecent > 70 then grade is ‘A’

If stud_pecent > 60 and <70 then grade is ‘B’ else give grade ‘C’.

Run SQL Command Line

```
SQL> declare
  2  rnum int := &rnum;
  3  m1 int := &mark1;
  4  m2 int := &mark2;
  5  p int;
  6  begin
  7  p := (m1 + m2)*100/2;
  8  if(p>70) then
  9  dbms_output.put_line('Rollno: '||rnum||' marks first: '||m1||' marks second '||m2||' Grade is A');
 10  elsif(p>60 and p<70) then
 11  dbms_output.put_line('Rollno: '||rnum||' marks first: '||m1||' marks second '||m2||' Grade is B');
 12  else
 13  dbms_output.put_line('Rollno: '||rnum||' marks first: '||m1||' marks second '||m2||' Grade is C');
 14  end if;
 15  end;
 16  /
Enter value for rnum: 4
old   2: rnum int := &rnum;
new   2: rnum int := 4;
Enter value for mark1: 87
old   3: m1 int := &mark1;
new   3: m1 int := 87;
Enter value for mark2: 88
old   4: m2 int := &mark2;
new   4: m2 int := 88;
Rollno: 4 marks first: 87 marks second 88 Grade is A

PL/SQL procedure successfully completed.
```

31. Write a PL/SQL block to book a ticket for a movie. The tickets are of two type's deluxe rows (D) and Ordinary rows (O). While booking the ticket the customer may ask 'D' or 'O' and number of ticket. For deluxe the rate is 350 and for ordinary 200. Find the total amount that the customer will pay and number of tickets (using case statement).

Run SQL Command Line

```
SQL> declare
  2  character char(10) := '&character';
  3  D varchar(5);
  4  O varchar(5);
  5  num int := &num;
  6  begin
  7  case
  8  when character='D' then dbms_output.put_line('You select the character: '||D||' Total Amount is: '||(num*350));
  9  when character='O' then dbms_output.put_line('You select the character: '||O||' Total Amount is: '||(num*200));
 10  end case;
 11  end;
 12  /
Enter value for character: O
old   2: character char(10) := '&character';
new   2: character char(10) := 'O';
Enter value for num: 6
old   5: num int := &num;
new   5: num int := 6;
You select the character:  Total Amount is: 1200

PL/SQL procedure successfully completed.
```