Ex.No.9 **PL/SQL STATEMENTS**

Date:09.05.2024

1)

SQL> set serveroutput on

SQL> DECLARE

2 message varchar2(20):='Hello, World!';

3 BEGIN

4 dbms\_output.put\_line(message);

5 END;

6 /

Hello, World!

PL/SQL procedure successfully completed.

2)

SQL> DECLARE

2 a integer:=0;

3 b integer:=20;

4 c integer;

5 f real;

6 BEGIN

7 c:=a+b;

8 dbms\_output.put\_line('Value of c: '||c);

9 f :=70.0/3.0;

10 dbms\_output.put\_line('Value of f: ' || f);

11 END;

12 /

Value of c: 20

Value of f: 23.33333333333333333333333333333333333333

PL/SQL procedure successfully completed.

3)

SQL> DECLARE

2 n number;

3 BEGIN

4 n:=&n;

5 IF n>0 THEN

6 Dbms\_output.put\_line('Given number is Greater than ZERO');

7 END IF;

8 END;

9 /

Enter value for n: 6

old 4: n:=&n;

new 4: n:=6;

Given number is Greater than ZERO

PL/SQL procedure successfully completed.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

4)

SQL> DECLARE

2 n number;

3 BEGIN

4 n:=&n;

5 IF n>0 THEN

6 Dbms\_output.put\_line('Given number is Greater than ZERO');

7 ELSE

8 Dbms\_output.put\_line('Given number is less than ZERO');

9 END IF;

10 END;

11 /

Enter value for n: -2

old 4: n:=&n;

new 4: n:=-2;

Given number is less than ZERO

PL/SQL procedure successfully completed.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

5)

SQL> DECLARE

2 n number;

3 BEGIN

4 n:=&n;

5 IF n > 0 THEN

6 Dbms\_output.put\_line('Given number is Greater than ZERO');

7 ELSE

8 IF n = 0 THEN

9 Dbms\_output.put\_line('Given number is Equal to ZERO');

10 ELSE

11 Dbms\_output.put\_line('Given number is Less than ZERO');

12 END IF;

13 END IF;

14 END;

15 /

Enter value for n: 6

old 4: n:=&n;

new 4: n:=6;

Given number is Greater than ZERO

PL/SQL procedure successfully completed.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

6)

SQL> DECLARE

2 n number;

3 BEGIN

4 n:=&n;

5 IF n>0 THEN

6 Dbms\_output.put\_line('Given number is Greater then ZERO');

7 ELSIF n = 0 THEN

8 Dbms\_output.put\_line('Given number is Equal to ZERO');

9 ELSE

10 Dbms\_output.put\_line('Given number is Less than ZERO');

11 END IF;

12 END;

13 /

Enter value for n: 7

old 4: n:=&n;

new 4: n:=7;

Given number is Greater then ZERO

PL/SQL procedure successfully completed.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

7)

SQL> DECLARE

2 grade CHAR(1);

3 BEGIN

4 grade:='B';

5 IF grade='A'THEN

6 DBMS\_OUTPUT.PUT\_LINE('Exellent');

7 ELSIF grade = 'B' THEN

8 DBMS\_OUTPUT.PUT\_LINE('Very Good');

9 ELSIF grade = 'C' THEN

10 DBMS\_OUTPUT.PUT\_LINE('Good');

11 ELSIF grade = 'D' THEN

12 DBMS\_OUTPUT. PUT\_LINE('Fair');

13 ELSIF grade = 'F' THEN

14 DBMS\_OUTPUT.PUT\_LINE('Poor');

15 ELSE

16 DBMS\_OUTPUT.PUT\_LINE('No such grade');

17 END IF;

18 END;

19 /

Very Good

PL/SQL procedure successfully completed.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

8)

SQL> DECLARE

2 grade CHAR(1);

3 BEGIN

4 grade := 'B';

5 CASE grade

6 WHEN 'A' THEN DBMS\_OUTPUT.PUT\_LINE('Excellent');

7 WHEN 'B' THEN DBMS\_OUTPUT.PUT\_LINE('Very Good');

8 WHEN 'C' THEN DBMS\_OUTPUT.PUT\_LINE('Good');

9 WHEN 'D' THEN DBMS\_OUTPUT.PUT\_LINE('Fair');

10 WHEN 'F' THEN DBMS\_OUTPUT.PUT\_LINE('Poor');

11 ELSE DBMS\_OUTPUT.PUT\_LINE('No such grade');

12 END CASE;

13 END;

14 /

Very Good

PL/SQL procedure successfully completed.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

9)

SQL> DECLARE

2 grade CHAR(1);

3 BEGIN

4 grade := 'B';

5 CASE

6 WHEN grade = 'A' THEN DBMS\_OUTPUT.PUT\_LINE('Excellent');

7 WHEN grade = 'B' THEN DBMS\_OUTPUT.PUT\_LINE('Very Good');

8 WHEN grade = 'C' THEN DBMS\_OUTPUT.PUT\_LINE('Good');

9 WHEN grade = 'D' THEN DBMS\_OUTPUT.PUT\_LINE('Fair');

10 WHEN grade = 'F' THEN DBMS\_OUTPUT.PUT\_LINE('Poor');

11 ELSE DBMS\_OUTPUT.PUT\_LINE('No such grade');

12 END CASE;

13 END;

14 /

Very Good

PL/SQL procedure successfully completed.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

10)

SQL> DECLARE

2 grade CHAR(1);

3 BEGIN

4 grade := 'B';

5 CASE

6 WHEN grade = 'A' THEN DBMS\_OUTPUT.PUT\_LINE('Excellent');

7 WHEN grade = 'B' THEN DBMS\_OUTPUT.PUT\_LINE('Very Good');

8 WHEN grade = 'C' THEN DBMS\_OUTPUT.PUT\_LINE('Good');

9 WHEN grade = 'D' THEN DBMS\_OUTPUT.PUT\_LINE('Fair');

10 WHEN grade = 'F' THEN DBMS\_OUTPUT.PUT\_LINE('Poor');

11 END CASE;

12 EXCEPTION

13 WHEN CASE\_NOT\_FOUND THEN

14 DBMS\_OUTPUT.PUT\_LINE('No such grade');

15 END;

16 /

Very Good

PL/SQL procedure successfully completed.

11)

SQL> declare

total number:=0;

begin

loop

total:=total+1;

exit when total>=5;

end loop;

dbms\_output.put\_line('total:'||total);

end;

/

total:5

PL/SQL procedure successfully completed.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

12)

SQL> Declare

i number:=0;

j number:=0;

begin

while i<=100

Loop

j := j+i;

i := i+2;

end loop;

dbms\_output.put\_line('the value of j is' ||j);

end;

/

the value of j is2550

PL/SQL procedure successfully completed.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

13)

SQL> DECLARE

A NUMBER;

I NUMBER :=1;

BEGIN

A:=10;

WHILE I<A LOOP

DBMS\_OUTPUT.PUT\_LINE('VALUE :'||I);

I:=I+1;

END LOOP;

END;

/

VALUE :1

VALUE :2

VALUE :3

VALUE :4

VALUE :5

VALUE :6

VALUE :7

VALUE :8

VALUE :9

PL/SQL procedure successfully completed.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

14)

SQL> BEGIN

FOR i IN 1..3

LOOP

DBMS\_OUTPUT.PUT\_LINE (TO\_CHAR(i));

END LOOP;

END;

/

1

2

3

PL/SQL procedure successfully completed.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

15)

SQL> BEGIN

FOR i IN REVERSE 1..3

LOOP

DBMS\_OUTPUT.PUT\_LINE (TO\_CHAR(i));

END LOOP;

END;

3

2

1

PL/SQL procedure successfully completed.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

16)

SQL> DECLARE

p VARCHAR2(30);

n PLS\_INTEGER := 37;

BEGIN

FOR j in 2..ROUND(SQRT(n))

LOOP

IF n MOD j = 0 THEN

p := ' is not a prime number';

GOTO print\_now;

END IF;

END LOOP;

p := ' is a prime number';

<<print\_now>>

DBMS\_OUTPUT.PUT\_LINE(TO\_CHAR(n) || p);

END;

/

37 is a prime number

PL/SQL procedure successfully completed.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

17)

SQL> DECLARE

v\_last\_name VARCHAR2(15);

v\_emp\_id NUMBER(6) := 1;

BEGIN

<<get\_name>>

BEGIN

SELECT ename into v\_last\_name from empl where eid=v\_emp\_id;

dbms\_output.put\_line(v\_last\_name);

v\_emp\_id:=v\_emp\_id+2;

if v\_emp\_id<=15 then

goto get\_name;

end if;

exception

when no\_data\_found then

dbms\_output.put\_line('No instructor found with ID'||v\_emp\_id);

end;

end;

/

Babu

Devi

Sabari

No instructor found with ID7

PL/SQL procedure successfully completed.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

18)

SQL>declare

n\_num number := 1;

begin

loop

dbms\_output.put(n\_num||', ');

n\_num := n\_num + 1;

exit when n\_num > 5;

end loop;

dbms\_output.put\_line('Final: '||n\_num);

end;

/

1, 2, 3, 4, 5, Final: 6

PL/SQL procedure successfully completed.

**19)**Factorial:

SQL> declare

n number:=5;

factorial number:=1;

begin

for i in 1..n loop

factorial:=factorial\*i;

end loop;

dbms\_output.put\_line('Factorial of '|| n ||' is '||factorial);

end;

/

Factorial of 5 is 120

PL/SQL procedure successfully completed.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20) Prime Number Generation:

SQL> declare

v\_limit number := 15;

isprime boolean;

begin

for i in 2..v\_limit loop

isprime := true;

for j in 2..sqrt(i) loop

if mod(i, j) = 0 then

isprime := false;

exit;

end if;

end loop;

if isprime then

dbms\_output.put\_line(i || ' is a prime number');

end if;

end loop;

end;

/

2 is a prime number

3 is a prime number

5 is a prime number

7 is a prime number

11 is a prime number

13 is a prime number

PL/SQL procedure successfully completed.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_21)Fibanacci Series:

SQL> declare

limit number:=10;

a number:=0;

b number:=1;

c number;

begin

dbms\_output.put\_line('Fibbonacci Series: ');

dbms\_output.put\_line(a);

dbms\_output.put\_line(b);

for i in 3..limit loop

c:=a+b;

dbms\_output.put\_line(c);

a:=b;

b:=c;

end loop;

end;

/

Fibbonacci Series:

0

1

1

2

3

5

8

13

21

34

PL/SQL procedure successfully completed.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

22)**checking palindrome**

SQL> declare

original varchar(20) := 'malayalam';

result varchar(100);

begin

for i in reverse 1..length(original) loop

result := result || substr(original, i, 1);

end loop;

if original = result then

dbms\_output.put\_line(original || ' is palindrome');

else

dbms\_output.put\_line(original || ' is not a palindrome');

end if;

end;

/

malayalam is palindrome

PL/SQL procedure successfully completed.

**23)Swap Two numbers:**

SQL> declare

num1 number := 10;

num2 number := 20;

temp number;

begin

dbms\_output.put\_line('Before Swapping:');

dbms\_output.put\_line('Number 1: ' || num1);

dbms\_output.put\_line('Number 2: ' || num2);

temp := num1;

num1 := num2;

num2 := temp;

dbms\_output.put\_line('After Swapping:');

dbms\_output.put\_line('Number 1: ' || num1);

dbms\_output.put\_line('Number 2: ' || num2);

end;

/

Before Swapping:

Number 1: 10

Number 2: 20

After Swapping:

Number 1: 20

Number 2: 10

PL/SQL procedure successfully completed.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

24)

SQL> declare

v\_eno empdet.eno%type;

v\_ename empdet.ename%type;

v\_deptno empdet.deptno%type;

v\_basic empdet.basic%type;

v\_hra empdet.HRA%type;

v\_da empdet.DA%type;

v\_pf empdet.PF%type;

v\_netpay empdet.netpay%type;

begin

v\_eno:=&v\_eno;

v\_ename:='&v\_ename';

v\_deptno:=&v\_deptno;

v\_basic:=&v\_basic;

v\_hra:=(v\_basic\*50)/100;

v\_da:=(v\_basic\*20)/100;

v\_pf:=(v\_basic\*7)/100;

v\_netpay:=v\_basic+v\_hra+v\_da-v\_pf;

insert into empdet(eno,ename,deptno,basic,HRA,DA,PF,netpay) values(v\_eno,v\_ename,v\_deptno,v\_basic,v\_hra,v\_da,v\_pf,v\_netpay);

dbms\_output.put\_line('Row inserted successfully.');

exception

when others then

dbms\_output.put\_line('Error:'|| SQLERRM);

end;

/

Enter value for v\_eno: 1

old 11: v\_eno:=&v\_eno;

new 11: v\_eno:=1;

Enter value for v\_ename: babu

old 12: v\_ename:='&v\_ename';

new 12: v\_ename:='babu';

Enter value for v\_deptno: 3

old 13: v\_deptno:=&v\_deptno;

new 13: v\_deptno:=3;

Enter value for v\_basic: 4000

old 14: v\_basic:=&v\_basic;

new 14: v\_basic:=4000;

Row inserted successfully.

PL/SQL procedure successfully completed.

25) PL/SQL block for updating multiple rows into a table.

**Update a single row:**

SQL> declare

2 new\_name varchar(15):='Mahi';

3 begin

4 update emp set name=new\_name where eid=3;

5 commit;

6 end;

7 /

PL/SQL procedure successfully completed.

SQL> select\* from emp;

EID NAME SALARY DNO

---------- -------------------- ---------- ----------

1 Babu 70000 3

2 Arasu 60000 4

3 Dhoni 60000 2

4 Virat 70000 1

5 Amal 50000 3

6 Muthu 55000 4

SQL> select \* from emp;

EID NAME SALARY DNO

---------- -------------------- ---------- ----------

1 Babu 70000 3

2 Arasu 60000 4

3 Mahi 60000 2

4 Virat 70000 1

5 Amal 50000 3

6 Muthu 55000 4

6 rows selected.

**Upadate a Multiple Row:**

SQL> declare

2 new\_salary number:=80000;

3 begin

4 update emp set salary=new\_salary where salary>55000;

5 commit;

6 end;

7 /

PL/SQL procedure successfully completed.

SQL> select \* from emp;

EID NAME SALARY DNO

---------- -------------------- ---------- ----------

1 Babu 80000 3

2 Arasu 80000 4

3 Mahi 80000 2

4 Virat 80000 1

5 Amal 50000 3

6 Muthu 55000 4

6 rows selected.

Result:

To PL/SQL statements in various programs are executed successfully.