**6. Queries on Working with Index, Sequence, Synonyms.**

**Syntax:**

create index indexname om tablename(columnname) ;

SQL> create index ind on stu1(upper(sname));

Index created.

**Syntax:**

alter index old\_name rename to new\_name;

SQL> alter index ind rename to index1;

Index altered.

**Syntax:**

drop index name;

SQL> drop index index1;

Index dropped.

SQL> create sequence s5 start with 1 increment by 1;

Sequence created.

SQL> insert into student values(s5.nextval,'rd',24);

1 row created.

SQL> insert into student values(s5.nextval,'rani',50);

1 row created.

SQL> insert into student values(s5.nextval,'raju',48);

1. row created.

SQL> select \* from student;

ROLLNO NAME MARKS

---------- -------------------- ---------- 1 rd 24

1. rani 50 3 raju 48 SQL> alter sequence s5 maxvalue 200;

Sequence altered.

SQL> select \* from user\_sequences;

SEQUENCE\_NAME MIN\_VALUE MAX\_VALUE INCREMENT\_BY C O CACHE\_SIZE

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

LAST\_NUMBER

-----------

S5 1 200 1 N N 20 4

SQL> drop sequence s5;

Sequence dropped.

SYNONYM:

Create synonym stu for student;

Drop synonym stu;

# Write a PL/SQL Code using Basic Variables and Usage of Assignment Operation

**Write a PL /SQL program to print hello world?**

SQL> set serveroutput on

SQL> begin

1. dbms\_output.put\_line('hello world');
2. end;
3. /

hello world

PL/SQL procedure successfully completed.

**Write a PL/SQL program to add two numbers?**

SQL> declare

1. a integer;
2. b integer;
3. c integer;
4. begin
5. a:=2;
6. b:=5;
7. c:=a+b;
8. dbms\_output.put\_line('sum of '||a||'and'||b||' is '||c);
9. end; 11 /

sum of 2and5 is 7

PL/SQL procedure successfully completed.

**Write PL/SQL program to print details of student?**

SQL> declare

1. roll integer;
2. name varchar(20);
3. age integer;
4. begin
5. roll:=5;
6. name:='xyzsa';
7. age:=20;
8. dbms\_output.put\_line('NAME OF STUDENT:'||name);
9. dbms\_output.put\_line('ROLL NUMBER OF STUDENT:'||roll);
10. dbms\_output.put\_line('AGE OF THE STUDENT:'||age); 12 end;

13 /

NAME OF STUDENT:xyzsa

ROLL NUMBER OF STUDENT:5

AGE OF THE STUDENT:20

PL/SQL procedure successfully completed.

**(OR)**

SQL> declare

1. ro integer;
2. name varchar(20);
3. age integer;
4. begin
5. ro:=&ro;
6. name:=&name;
7. age:=&age;
8. dbms\_output.put\_line('rollno:'||ro);
9. dbms\_output.put\_line('name:'||name);
10. dbms\_output.put\_line('age:'||age);
11. end;
12. /

Enter value for ro: 45 old 6: ro:=&ro; new 6: ro:=45; Enter value for name: 'afg' old 7: name:=&name; new 7: name:='afg'; Enter value for age: 5 old 8: age:=&age; new 8: age:=5; rollno:45 name:afg

age:5

PL/SQL procedure successfully completed.

**9. Write a PL/SQL Code to Bind and Substitute variables in PL/SQL.**

**BIND VARIABLE**

SQL> variable a number

SQL> begin

1. :a:=1;
2. end;
3. /

PL/SQL procedure successfully completed.

SQL> print a;

A

---------- 1

SQL> exec:a:=2;

PL/SQL procedure successfully completed.

SQL> print a;

A

---------- 2

**SUBSTITUTION VARIABLE**

SQL> define name='ravi'; SQL> select '&&name' from dual; old 1: select '&&name' from dual

new 1: select 'ravi' from dual

'RAV ---- ravi

SQL> undefine name;

SQL> select '&&name' from dual; Enter value for name: raja old 1: select '&&name' from dual new 1: select 'raja' from dual

'RAJ

---- raja

**10.Write PL/SQL block using sql and control structures**

**Write PL/SQL program to find given number is even or odd**

SQL> declare

1. n int;
2. begin
3. n:=&n;
4. if mod(n,2)=0 then
5. dbms\_output.put\_line(n||' is EVEN NUMBER');
6. else
7. dbms\_output.put\_line(n||' is ODD NUMBER');
8. end if;
9. end;
10. /

Enter value for n: 88

old 4: n:=&n;

new 4: n:=88;

88 is EVEN NUMBER

PL/SQL procedure successfully completed.

**Write PL/SQL program to find biggest of two numbers**

SQL> declare

1. a int;
2. b int;
3. begin
4. a:=&a;
5. b:=&b;
6. if (a>b) then
7. dbms\_output.put\_line(a||' is the biggest number'); 9 else
8. dbms\_output.put\_line(b||' is the biggest number');
9. end if;
10. end;
11. /

Enter value for a: 89 old 5: a:=&a; new 5: a:=89; Enter value for b: 45 old 6: b:=&b; new 6: b:=45;

89 is the biggest number

PL/SQL procedure successfully completed.

**Write a PL/SQL program to take marks as input and print the status?**

SQL> declare

1. marks integer:=&marks;
2. begin
3. if(marks>=75)then
4. dbms\_output.put\_line('DISTINCTION');
5. elsif(marks>=60 and marks<75)then
6. dbms\_output.put\_line('FIRST CLASS');
7. elsif(marks>=50 and marks<60)then
8. dbms\_output.put\_line(''SECOND CLASS');
9. ELSE
10. dbms\_output.put\_line('FAIL');
11. end if;
12. end;
13. /

Enter value for marks: 67 old 2: marks integer:=&marks; new 2: marks integer:=67;

FIRST CLASS

PL/SQL procedure successfully completed.

SQL> /

Enter value for marks: 98 old 2: marks integer:=&marks; new 2: marks integer:=98;

DISTINCTION

PL/SQL procedure successfully completed.

**CASE SELECTOR**

**Write a PL/SQL program to take input grade and print status?**

SQL> declare

1. grade char(1):='&grade';
2. begin
3. case grade
4. when 'A' then dbms\_output.put\_line('EXCELLENT');
5. when 'B' then dbms\_output.put\_line('GOOD');
6. when 'C' then dbms\_output.put\_line('AVERAGE');
7. when 'D' then dbms\_output.put\_line('BAD');
8. end case;
9. end;
10. /

Enter value for grade: A old 2: grade char(1):='&grade';

new 2: grade char(1):='A';

EXCELLENT

SQL> /

Enter value for grade: D old 2: grade char(1):='&grade';

new 2: grade char(1):='D';

BAD

**NESTED IF**

**Write a PL/SQL program to print biggest of three numbers?**

SQL>declare a int:=&a; b int:=&b; c int:=&c; begin if(a>b)then if(a>c)then

dbms\_output.put\_line(a||'is the biggest'); else

dbms\_output.put\_line(c||'is the biggest'); end if; else if(b>c)then

dbms\_output.put\_line(b||' is the biggest'); else

dbms\_output.put\_line(c||' is the biggest'); end if; end if;

end;

/

Enter value for a: 8 old 2: a int:=&a; new 2: a int:=8; Enter value for b: 47 old 3: b int:=&b; new 3: b int:=47; Enter value for c: 16 old 4: c int:=&c; new 4: c int:=16;

47 is the biggest

PL/SQL procedure successfully completed.

**LOOPS**

**Write a PL/SQL program to print sequence of n numbers using simple loop**

SQL> declare

1. a integer;
2. n integer;
3. begin
4. a:=1;
5. n:=&n;
6. loop
7. dbms\_output.put\_line(a);
8. a:=a+1;
9. exit when a>n;
10. end loop;
11. end;
12. /

Enter value for n: 8 old 6: n:=&n; new 6: n:=8;

1

2

3

4

5

6

7

8

PL/SQL procedure successfully completed.

**Write a PL/SQL program to print sequence of n numbers using WHILE LOOP?**

SQL> declare

1. a int:=1;
2. n int:=&n;
3. begin
4. while(a<n)loop
5. dbms\_output.put\_line(a);
6. a:=a+1;
7. end loop;
8. end;
9. /

Enter value for n: 6 old 3: n int:=&n; new 3: n int:=6;

1

2

3

4

5

PL/SQL procedure successfully completed.

**Write a PL/SQL program to print sequence of n numbers using FOR LOOP?**

SQL> declare

1. a int:=1;
2. n int:=&n;
3. brgin
4. for a in 1..n
5. loop
6. dbms\_output.put\_line(a);
7. end loop;
8. end;
9. /

Enter value for n: 5 old 3: n int:=&n; new 3: n int:=5;

1

2

3

4

5

PL/SQL procedure successfully completed.

**WRITE a PL/SQL program to print a multiplication table?**

SQL> declare

1. n int:=&n;
2. k int:=&k;
3. I int:=1;
4. begin
5. while(i<=k)loop
6. dbms\_output.put\_line(n||' x '||i||' = '||(n\*i));
7. i:=i+1;
8. end loop;
9. end;
10. /

SQL> /

Enter value for n: 2 old 2: n int:=&n; new 2: n int:=2; Enter value for k: 10 old 3: k int:=&k; new 3: k int:=10;

2 x 1 = 2

2 x 2 = 4

2 x 3 = 6

2 x 4 = 8

2 x 5 = 10

2 x 6 = 12

2 x 7 = 14

2 x 8 = 16

2 x 9 = 18

2 x 10 = 20

PL/SQL procedure successfully completed.

**Write a PL/SQL program to find the factorial of a given number?**

SQL> declare

1. n int:=&n;
2. i int:=1;
3. f int:=1;
4. begin
5. while(i<=n)loop
6. f:=f\*i;
7. i:=i+1;
8. end loop;
9. dbms\_output.put\_line(n||'factorial = '||f);
10. end;
11. /

Enter value for n: 5 old 2: n int:=&n; new 2: n int:=5;

5factorial = 120

PL/SQL procedure successfully completed.

**Write a PL/SQL program to find given number is a prime or not?**

SQL> declare

1. n int:=&n;
2. i int:=1;
3. c int:=0;
4. begin
5. while(i<=n)
6. loop
7. if(mod(n,i)=0)then
8. c:=c+1; 10 end if;
9. i:=i+1;
10. end loop;
11. if(c=2)then
12. dbms\_output.put\_line(n||' is a prime number');
13. else
14. dbms\_output.put\_line(n||' is not a prime number');
15. end if;
16. end;
17. /

Enter value for n: 7 old 2: n int:=&n; new 2: n int:=7;

1. is a prime number

PL/SQL procedure successfully completed.

SQL> /

Enter value for n: 8 old 2: n int:=&n; new 2: n int:=8;

1. is not a prime number

PL/SQL procedure successfully completed.

**Write a PL/SQL program to check given number is Armstrong or not?**

SQL>declare n number:=&n; s number:=0; r number; m number;

begin

m := n; while n>0 loop r := mod(n , 10); s := s + power(r , 3); n := trunc(n / 10); end loop; if m = s then

dbms\_output.put\_line(m||' is the armstrong number'); else

dbms\_output.put\_line(m||' is not armstrong number'); end if; end;

/

Enter value for n: 153 old 2: n number:=&n; new 2: n number:=153; 153 is the armstrong number

PL/SQL procedure successfully completed.