	Roll No:1403210009Date	Page No
Practical Name		Practical No.

### **Prerequisite**:

#### PL/SQL

PL/SQL stands for Procedural Language extensions to the Structured Query Language (SQL). SQL is a powerful language for both querying and updating data in relational databases.

#### PL/SQL Procedure

The PL/SQL stored procedure or simply a procedure is a PL/SQL block which performs one or more specific tasks. It is just like procedures in other programming languages.

The procedure contains a header and a body.

- **Header:** The header contains the name of the procedure and the parameters or variables passed to the procedure.
- **Body:** The body contains a declaration section, execution section and exception section similar to a general PL/SQL block.

#### **PL/SQL Create Procedure**

#### **Syntax for creating procedure:**

- 1. CREATE [OR REPLACE] PROCEDURE procedure\_name
- 2. [ (parameter [,parameter]) ]
- 3. IS
- 4. [declaration\_section]
- 5. BEGIN
- 6. executable\_section
- 7. [EXCEPTION
- 8. exception\_section]
- END [procedure\_name];

### PL/SQL code block for Addition:

```
SQL> set serveroutput ON;
Declare
a number(3) := &a;
b number(3) := &b;
c number(3);
Begin
c := a+b;
Dbms output.put line(c);
SQL> 2 3 4 5 6 7 8 9 /
Enter value for a: 45
old 2: a number(3) := &a;
new 2: a number(3) := 45;
Enter value for b: 88
old 3: b number(3) := &b;
new 3: b number(3) := 88;
PL/SQL procedure successfully completed
```

### PL/SQL code block for Multiplication:

```
SQL>
set serveroutput ON;
Declare
a number(3) := &a;
b number(3) := &b;
c number(3) := &c;
d number(3);
Begin
d := a*b*c;
Dbms_output.put_line(d);
end;SQL>SQL> 2 3 4 5 6 7 8 9
Enter value for a: 2
old 2: a number(3) := &a;
new 2: a number(3) := 2;
Enter value for b: 3
old 3: b number(3) := &b;
new 3: b number(3) := 3;
Enter value for c: 4
old 4: c number(3) := &c;
new 4: c number(3) := 4;
```

### PL/SQL code block for finding factorial of a number:

```
SQL>
set serveroutput ON;
Declare
a number(3) := 5;
b number(10);
inumber(3);
Begin
b := 1;
for i in 1 .. a
loop
b := b*i;
end loop;
Dbms_output.put_line(b);
SQL>SQL>SQL> 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
21 /
120
```

# PL/SQL code block for finding fibonacci series:

```
SQL> set serveroutput ON;
Declare
a NUMBER(3):=0;
b NUMBER(3):=1;
c NUMBER(3);
cnt NUMBER(3):=8;
i NUMBER(3);
Begin
dbms_output.put_line(a);
dbms_output.put_line(b);
for i in 1..cnt
loop
c:=a+b;
a:=b;
b:=c;
dbms_output.put_line(c);
end loop;
end:
SQL> 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
0
1
1
2
3
5
8
13
21
34
PL/SQL procedure successfully completed.
```

## PL/SQL code block for finding reverse of a list:

	Roll No:1403210009	DatePa	ge No
Practical Name		Practic	al No

```
SQL>
set serveroutput ON;
Declare
a varchar(20) := 'Akansha';
b varchar(20);
Inumber(20);
i number(3);
j number(3);
Begin
I:=length(a);
j :=l;
for i in 1 .. I
loop
 b := concat(b,substr(a,j,1));
j := j-1;
end loop;
Dbms_output.put_line(b);
SQL>SQL> 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22
23 /
ahsnakA
PL/SQL procedure successfully completed.
SQL>
```