Experiment 8
#8.1

```
create table employee(empid varchar2(20) primary key, empname varchar2(20), job
varchar2(10), manager varchar2(20), deptno varchar2(20), hiredate date, Comm number(5),  sal
number(7));
Insert into employee values('E0001', 'Abey', 'Tester', 'E0004', 'D004', '15-DEC-2012', 525,
30000);
Insert into employee values('E0002', 'Jesto', 'Analyst', 'E0001', 'D002', '24-apr-2011', 683,
25000);
Insert into employee values('E0003', 'Adarsh', 'Clerk', 'E0004', 'D004', '10-jan-2013', 525,
15000);
Insert into employee values('E0004', 'Kevin', 'Admin', 'E0005', 'D002', '10-oct-2013', 1050,
20000);
Insert into employee values('E0005', 'Bony', 'Manager',' ', 'D001', '11-apr-2011', 1000, 50000);
Insert into employee values('E0006', 'Manu', 'Supplier', 'E0001',  'D003', '19-jun-2013', 473,
5000);

DECLARE
id varchar2(6);
sal_incr number(5);

PROCEDURE salincrement(id IN varchar2, sal_incr IN number)
IS
BEGIN
update EMPLOYEE set sal=sal+sal_incr where empid=id;
END;
BEGIN
  id:='&Empid';
sal_incr:='&Increment';
salincrement(id, sal_incr);
END;
```

```
SQL> DECLARE
  2  id varchar2(6);
  3  sal_incr number(5);
  4
  5  PROCEDURE salincrement(id IN varchar2, sal_incr IN number)
  6  IS
  7  BEGIN
  8  update EMPLOYEE set sal=sal+sal_incr where empid=id;
  9  END;
 10  BEGIN
 11    id:='&Empid';
 12  sal_incr:='&Increment';
 13  salincrement(id, sal_incr);
 14  END;
 15  /
Enter value for empid: E0003
old  11:   id:='&Empid';
new  11:   id:='E0003';
Enter value for increment: 5000
old  12: sal_incr:='&Increment';
new  12: sal_incr:='5000';

PL/SQL procedure successfully completed.
```

#8.2

```
DECLARE
id varchar2(6);
PROCEDURE increment(id IN varchar2)
IS
BEGIN
update EMPLOYEE set comm=0.05*comm+comm where empid=id;
END;
BEGIN
  id:='&Empid';
increment(id);
END;
```

```
Commit complete.
SQL> DECLARE
  2  id varchar2(6);
  3  PROCEDURE increment(id IN varchar2)
  4  IS
  5  BEGIN
  6  update EMPLOYEE set comm=0.05*comm+comm where empid=id;
  7  END;
  8  BEGIN
  9    id:='&Empid';
 10  increment(id);
 11  END;
 12  /
Enter value for empid: E0004
old   9:   id:='&Empid';
new   9:   id:='E0004';

PL/SQL procedure successfully completed.

Commit complete.
```

#8.3

```
create table bank(username varchar2(20), accid number(4), balance number(6,2));
insert into bank values('Nihal'', 1234, 2000.5);
insert into bank values('Sarah', 1235, 1590.7);
insert into bank values('Neha', 1236, 2908.5);
insert into bank values('Neema', 1238, 2560.4);
insert into bank values('Noel', 1239, 1245.2);
DECLARE
id number(4);
wamt number(6,2);
balamt number(6,2);
bal number(6,2);
PROCEDURE withdraw(wamt IN number, id IN number)
IS
BEGIN
select balance into balamt from bank where id=accid;
bal:=balamt-wamt;
if(bal<1000) then
dbms_output.put_line('LOW BALANCE!!');
else
update bank set balance=bal where id=accid;
dbms_output.put_line('Money withdrawn! Balance: '|| bal);
end if;
END;
BEGIN
id:='&id';
wamt:='&wamt';
withdraw(wamt, id);
```

END;

```
SQL> DECLARE
  2  id number(4);
  3  wamt number(6,2);
  4  balamt number(6,2);
  5  bal number(6,2);
  6  PROCEDURE withdraw(wamt IN number, id IN number)
  7  IS
  8  BEGIN
  9  select balance into balamt from bank where id=accid;
 10  bal:=balamt-wamt;
 11  if(bal<1000) then
 12  dbms_output.put_line('LOW BALANCE!!');
 13  else
 14  update bank set balance=bal where id=accid;
 15  dbms_output.put_line('Money withdrawn! Balance: '|| bal);
 16  end if;
 17  END;
 18  BEGIN
 19  id:='&id';
 20  wamt:='&wamt';
 21  withdraw(wamt, id);
 22  END;
 23  /
Enter value for id: 1234
old  19: id:='&id';
new  19: id:='1234';
Enter value for wamt: 1000
old  20: wamt:='&wamt';
new  20: wamt:='1000';
Money withdrawn! Balance: 1000.5

PL/SQL procedure successfully completed.

Commit complete.
SQL>
```

#8.4

```
SQL> DECLARE
  2         num1 INTEGER;
  3       num2 INTEGER;
  4       t    INTEGER;
  5  BEGIN
  6       num1 := &m;
  7        num2 := &n;
  8     WHILE MOD(num2, num1) != 0 LOOP
  9           t := MOD(num2, num1);
 10
 11           num2 := num1;
 12
 13           num1 := t;
 14       END LOOP;
 15
 16       dbms_output.Put_line('GCD of '
 17                            ||num1
 18                            ||' and '
 19                            ||num2
 20                            ||' is '
 21                            ||num1);
 22  END;
 23  /
Enter value for m: 4
old   6:      num1 := &m;
new   6:      num1 := 4;
Enter value for n: 2
old   7:       num2 := &n;
new   7:       num2 := 2;
GCD of 2 and 4 is 2

PL/SQL procedure successfully completed.

Commit complete.
SQL>
```

#8.5

```
CREATE TABLE Student6 (RegNo INTEGER PRIMARY KEY,name VARCHAR2(20),marks NUMBER);

INSERT INTO Student6 VALUES (1, 'Alice', 85);

INSERT INTO Student6 VALUES (2, 'Rohan', 75);

INSERT INTO Student6 VALUES (4, 'Mat', 80);

INSERT INTO Student6 VALUES (5, 'Joe', 95);
```

```
SQL>
SQL> CREATE OR REPLACE PROCEDURE disp_avg_mark
  2  AS
  3    v_average NUMBER;
  4  BEGIN
  5    SELECT AVG(marks) INTO v_average FROM Student6;
  6    DBMS_OUTPUT.PUT_LINE('Average mark: ' || v_average);
  7  END;
  8  /

Procedure created.

SQL> BEGIN
  2     disp_avg_mark;
  3  END;
  4  /
Average mark: 83.75

PL/SQL procedure successfully completed.

Commit complete.
SQL> CREATE OR REPLACE PROCEDURE disp_mark (reg_no IN INTEGER)
  2  AS
  3    mark NUMBER;
  4  BEGIN
  5    SELECT marks INTO mark FROM Student6 WHERE RegNo = reg_no;
  6    DBMS_OUTPUT.PUT_LINE('Mark: ' || mark);
  7  END;
  8  /

Procedure created.

SQL>
SQL> BEGIN
  2    disp_mark(1);
  3  END;
  4  /
Mark: 85

PL/SQL procedure successfully completed.
```