PL/SQL

NOTE: For some of the programs which are based on tables, you may need to create related tables before execution of the respective programs.

Introduction(Basic Program):

SQL>Write a program to display welcome message.

```
BEGIN
DBMS_OUTPUT.PUT_LINE('HAI');
DBMS_OUTPUT.PUT_LINE('WELCOME');
DBMS_OUTPUT.PUT_LINE('PL/SQL PROGRAMS');
END;
```

OUTPUT:

```
HAI
WELCOME
PL/SQL PROGRAMS

PL/SQL procedure successfully completed
```

1) PL/SQL Code using Basic Variable, Anchored Declarations, and Usage of Assignment Operation

a) Write a program to find sum of two integer numbers.

SQL>

```
DECLARE
A NUMBER;
B NUMBER;
 C NUMBER;
 BEGIN
A:=100;
B:=200;
C:=A+B;
DBMS_OUTPUT.PUT_LINE('THE SUM OF TWO INTEGERS IS:  '||C);
END;
```

OUTPUT:

```
THE SUM OF TWO INTEGERS IS:  3
```

PL/SQL procedure successfully completed.

b) Write a program to accept empno,ename,sal & calculate bonus on the following condition 20% on ann_sal.

SQL>

```
      DECLARE
      EMPNO NUMBER;
      ENAME VARCHAR2(20);
      SAL NUMBER(7,2);
      ANU_SAL NUMBER(10,2);
      BONUS NUMBER(10,2);
      BEGIN
      EMPNO:=1234;
      ENAME:='Ravi';
      SAL:=18000;
      ANU_SAL:=SAL*12;
      BONUS:=ANU_SAL*20/100;
      DBMS_OUTPUT.PUT_LINE('EMPNO: '||EMPNO);
      DBMS_OUTPUT.PUT_LINE('ENAME: '||ENAME);
      DBMS_OUTPUT.PUT_LINE('SAL:  '||SAL);
      DBMS_OUTPUT.PUT_LINE('BONUS: '||BONUS);
     END;
```

OUTPUT:

```
          EMPNO: 1234
          ENAME: ravi

          SAL: 18000

          BONUS: 43200

Statement processed.
```

c) Write a program to accept product no,pname,quantity,price & calculate total,discount(20% on total),net bill.

SQL>

```
      DECLARE
      PRODNO NUMBER;
      PNAME VARCHAR2(20);
      QUAN NUMBER(3);
      PRICE NUMBER(7,2);
      TOTAL NUMBER(7,2);
      DISCOUNT NUMBER(7,2);
      NET NUMBER(7,2);
      BEGIN
      PRODNO:=1234;
      PNAME:='Chocolates';
      QUAN:=10;
      PRICE:=100;
      TOTAL:=QUAN*PRICE;
      DISCOUNT:=TOTAL*20/100;
```

```
NET:=TOTAL-DISCOUNT;
DBMS_OUTPUT.PUT_LINE('PRODNO: '||PRODNO);
DBMS_OUTPUT.PUT_LINE('PNAME: '||PNAME);
DBMS_OUTPUT.PUT_LINE('QUANTITY: '||QUAN);
DBMS_OUTPUT.PUT_LINE('PRICE: '||PRICE);
DBMS_OUTPUT.PUT_LINE('TOTAL: '||TOTAL);
DBMS_OUTPUT.PUT_LINE('DISCOUNT: '||DISCOUNT);
DBMS_OUTPUT.PUT_LINE('NET BALANCE: '||NET);
END;
```

OUTPUT:

```
PRODNO: 1234

PNAME: &Chocolates

QUANTITY: 10

PRICE: 100

TOTAL: 1000

DISCOUNT: 200

NET BALANCE: 800
```

2. Write a PL/SQL block using SQL and Control Structures in PL/SQL

a)Write a program to accept empno,sal,calculate bonus based on the following conditions

Salary	**Bonus**
>=10000	**20% on ann_sal**
>=5000&<10000	**15% on ann_sal**
>=3000&<5000	**12% on ann_sal**
>=1500&<3000	**10% on ann_sal**
>1500	**8% on ann_sal**

SQL>

```
DECLARE
  EMPNO NUMBER;
  SAL NUMBER(7,2);
  ANU_SAL NUMBER(7,2);
  BONUS NUMBER(7,2);
  BEGIN
  EMPNO:=1234;
  SAL:=8000;
  ANU_SAL:=SAL*12;
  IF SAL>=10000 THEN
  BONUS:=ANU_SAL*20/100;
  ELSIF SAL>=5000 AND SAL<10000 THEN
  BONUS:=ANU_SAL*15/100;
```

```
ELSIF SAL>=3000 AND SAL<5000 THEN
BONUS:=ANU_SAL*12/100;
ELSIF SAL>=1500 AND SAL<3000 THEN
BONUS:=ANU_SAL*10/100;
ELSE
BONUS:=ANU_SAL*8/100;
END IF;
DBMS_OUTPUT.PUT_LINE('EMPNO: '||EMPNO);
DBMS_OUTPUT.PUT_LINE('SAL: '||SAL);
DBMS_OUTPUT.PUT_LINE('ANU_SAL: '||ANU_SAL);
DBMS_OUTPUT.PUT_LINE('BONUS: '||BONUS);
END;
```

OUTPUT:

```
EMPNO: 1234
SAL: 8000

ANU_SAL: 96000

BONUS: 14400

Statement processed.
```

b)Write a Program to print numbers from 10-1.

SQL>

```
DECLARE
I NUMBER;
BEGIN
DBMS_OUTPUT.PUT_LINE('THE NUMBERS ARE');
FOR I IN REVERSE 1..10 LOOP
DBMS_OUTPUT.PUT_LINE(I);
END LOOP;
END;
```

OUTPUT:

```
THE NUMBERS ARE
10
9
8
7
6
5
4
3
2
1

PL/SQL procedure successfully completed.
```

c) Write a Program to accept a date & print next 7 days along with day.

SQL>

```
DECLARE
DA DATE;
I NUMBER;
BEGIN
DA:='10-04-2010';
FOR I IN 1..7 LOOP
DBMS_OUTPUT.PUT_LINE('THE DATE IS:'||DA);
DA:=DA+1;
END LOOP;
 END;
```

OUTPUT:

```
                    THE DATE IS:10/04/2010
THE DATE IS:10/05/2010

THE DATE IS:10/06/2010

THE DATE IS:10/07/2010

THE DATE IS:10/08/2010

THE DATE IS:10/09/2010

THE DATE IS:10/10/2010

Statement processed
```

d)Write a Program to display dept details

Note: Create a table with name DEPT and columns DEPTNO,DNAME,LOC with data inserted into it before running the below program and write ouptut according to the data inserted.

SQL>

```
DECLARE
CURSOR EC IS SELECT * FROM DEPT;
BEGIN
FOR V_EC IN EC
LOOP
DBMS_OUTPUT.PUT_LINE('DEPTNO='||V_EC.DNO);
DBMS_OUTPUT.PUT_LINE('DNAME='||V_EC.DNAME);
DBMS_OUTPUT.PUT_LINE('LOC='||V_EC.LOC);
END LOOP;
END;
```

Output:

```
DEPTNO=102
DNAME=Designing

LOC=Ongole

DEPTNO=101

DNAME=Development

LOC=Kakinada

DEPTNO=103

DNAME=Sales

LOC=Guntur

Statement processed.
```

e)Write a Program to increment all emp sal by 10% who are working in grade 2&3.

```
SQL>
        DECLARE
        CURSOR EC IS SELECT E.EMPNO,E.ENAME,E.SAL,S.GRADE FROM EMP
        E,SALGRADE S WHERE S.GRADE IN(2,3) AND E.SAL BETWEEN S.LOSAL AND
        S.HISAL;
        BEGIN
        FOR V_EC IN EC
        LOOP
        UPDATE EMP SET SAL=V_EC.SAL+V_EC.SAL*0.1 WHERE EMPNO=V_EC.EMPNO;
        END LOOP;
        END;
```

OUTPUT:

Before :

```
EMPNO ENAME          SAL     GRADE
---------- ---------- -------------------- ----------
    7654 MARTIN        1250         2
    7521 WARD          1250         2
    7934 MILLER         1300        2
    7499 ALLEN          1600        3
    7844 TURNER         1500        3
```

After:

```
 EMPNO ENAME            SAL      GRADE
---------- ---------- ---------- -------------------
      7654 MARTIN          1375           2
      7521 WARD            1375          2
      7499 ALLEN           1760          3
      7844 TURNER          1650          3
      7934 MILLER          1430          3
```

3. Write a PL/SQL Code using Cursors, Exceptions and Composite Data Types

a)wap to display emp details along with ann_sal & exp

SQL>

```
DECLARE
CURSOR C1 IS SELECT  * FROM EMP;
V_EC  C1%ROWTYPE;
ANN_SAL  NUMBER(8,2);
EXP NUMBER(8,2);
BEGIN
OPEN C1;
LOOP
FETCH C1 INTO V_EC;
EXIT WHEN C1%NOTFOUND;
ANN_SAL := V_EC.SAL*12;
EXP := MONTHS_BETWEEN(SYSDATE,V_EC.HIREDATE)/12;
DBMS_OUTPUT.PUT_LINE('EMPNO=' ||V_EC.EMPNO);
DBMS_OUTPUT.PUT_LINE('ANN_SAL=' ||ANN_SAL);
DBMS_OUTPUT.PUT_LINE('EXP=' ||EXP);
END LOOP;
CLOSE C1;
END;
```

OUTPUT:

```
empno=7839
ann_sal=60000
exp=30.37
empno=7698
ann_sal=34200
exp=30.91
empno=7782
ann_sal=29400
```

```
exp=30.81
empno=7566
ann_sal=35700
exp=30.99
empno=7654
ann_sal=15000
exp=30.5
empno=7499
ann_sal=19200
exp=31.11
empno=7844
ann_sal=18000
exp=30.56
empno=7900
ann_sal=11400
exp=30.32
empno=7521
ann_sal=15000
exp=31.1
empno=7902
ann_sal=36000
exp=30.32
empno=7369
ann_sal=9600
exp=31.28
empno=7788
ann_sal=36000
exp=29.31
empno=7876
ann_sal=13200
exp=29.21
empno=7934
ann_sal=15600
exp=30.18

PL/SQL procedure successfully completed.
```

b)Write a Program to calc bonus for all emps insert into bonus table

```
SQL>  CREATE TABLE BONUS1(EMPNO NUMBER(5) PRIMARY KEY,BONUS_AMT
      NUMBER(10,3),ADD_AMT NUMBER(10,3),ISS_DATE DATE);
```

SQL>

```
       DECLARE
       CURSOR EC IS SELECT EMPNO,SAL FROM EMP;
       V_EC EC%ROWTYPE;
       ANN_SAL NUMBER(10,2);
       B BONUS1%ROWTYPE;
       BEGIN
       OPEN EC;
```

```
 LOOP
 FETCH EC INTO V_EC;
 EXIT WHEN EC%NOTFOUND;
 ANN_SAL :=V_EC.SAL*12;
 B.BONUS_AMT := ANN_SAL*0.2;
 INSERT INTO BONUS1(EMPNO,BONUS_AMT,ADD_AMT,ISS_DATE)
VALUES(V_EC.EMPNO,B.BONUS_AMT,1000,SYSDATE
 END LOOP;
 CLOSE EC;
 END;
```

OUTPUT:

```
SQL>  SELECT *FROM BONUS1;

          EMPNO  BONUS_AMT    ADD_AMT ISS_DATE
          ------------------------------ ---------- ---------- ---------
          7839     12000         1000           29-MAR-12
           7698     6840         1000           29-MAR-12
          7782      5880         1000           29-MAR-12
          7566      7140         1000           29-MAR-12
          7654      3300         1000           29-MAR-12
          7499      4224         1000           29-MAR-12
          7844      3960         1000           29-MAR-12
          7900      2280         1000           29-MAR-12
          7521      3300         1000           29-MAR-12
          7902      7200         1000           29-MAR-12
          7369      1920         1000           29-MAR-12
```

c)Write a Program to display empno,ename,job,sal,deptno,dname,loc,grade of all managers.

SQL>

```
DECLARE
CURSOR EC IS SELECT E.EMPNO,E.JOB,E.SAL,D.DEPTNO,D.LOC,S.GRADE FROM
EMP E,SALGRADE S,DEPT D WHERE E.JOB='MANAGER' AND
E.DEPTNO=D.DEPTNO AND E.SAL BETWEEN S.LOSAL AND S.HISAL;
V_EC EC%ROWTYPE;
BEGIN
OPEN EC;
LOOP
FETCH EC INTO V_EC;
EXIT WHEN EC%NOTFOUND;
DBMS_OUTPUT.PUT_LINE(V_EC.EMPNO);
DBMS_OUTPUT.PUT_LINE(V_EC.JOB);
DBMS_OUTPUT.PUT_LINE(V_EC.SAL);
DBMS_OUTPUT.PUT_LINE(V_EC.DEPTNO);
DBMS_OUTPUT.PUT_LINE(V_EC.LOC);
DBMS_OUTPUT.PUT_LINE(V_EC.GRADE);
END LOOP;
CLOSE EC;
END;
```

OUTPUT:

```
7698
MANAGER
2850
30
CHICAGO
4
7782
MANAGER
2450
10
NEW YORK
4
7566
MANAGER
2975

20
DALLAS
4

PL/SQL procedure successfully completed.
```

4. Write a PL/SQL Code using Procedures, Functions, and Packages FORMS

a)Write a program with Procedure to print Min of two numbers in PL/SQL

sql>

```
DECLARE
a number;
b number;
c number;
CREATE OR REPLACE PROCEDURE min(x IN number, y IN number, z OUT
number)
IS
BEGIN
IF x<y THEN
z:=x;
ELSE
z:=y;
END IF;
END;
BEGIN
a:=25;
b:=40;
```

```
minimum(a,b,c);
dbms_output.put_line(c);
END;
```

Output:

```
        40
```

b)Write a program with Procedure to print Square of a number in PL/SQL.
Sql>

```
CREATE OR REPLACE PROCEDURE square(x IN OUT number)
IS
BEGIN
x:=x*x;
END;
DECLARE
a number;
BEGIN
a:=5;
square(a);
dbms_output.put_line(‘Square of 5 is: ’||a);
END;
```

Output:

```
Statement Processed
Procedure Created
Square of 5 is 25
```

c)Write a function to find maximum of two numebrs in PL/SQL
sql>

```
DECLARE
a number;
b number;
c number;
FUNCTION max(x in number, y in number)
RETURN number
IS
z number;
BEGIN
IF x>y THEN
z:=x;
ELSE
z:=y;
END IF;
RETURN z;
END Max;
```

```
BEGIN
a:=20;
b:=40;
c:=max(a, b);
dbms_output.put_line(‘Maximum of (20, 40):’||c);
END;
```

Output:

```
Maximum of (20, 40): 40
```

d)Write a function to check whether the given number is palindrome or not in PL/SQL

```
DECLARE
x number;
y number;
z number;
FUNCTION palindrome(a in out number)
RETURN number IS
temp number;
rem number;
BEGIN
temp:=0;
m:=n;
WHILE(n>0)
LOOP
rem:=mod(n, 10);
temp:=(temp*10)+rem;
n:=n/10;
END LOOP;
RETURN m;
END;
BEGIN
x:=1234;
y:=x;
z:=palindrome(x)
IF (z=y) THEN
dbms_output.put_line(‘palindrome);
ELSE
dbms_output.put_line(‘not palindrome’);
END IF;
END;
```

Output:

```
palindrome
```