**Practical 4**

**FUNCTIONS:**

1. Write a PL/SQL block to find the factorial of a number without recursion using function.

**Code:**

**SET SERVEROUTPUT ON;**

**CREATE OR REPLACE FUNCTION FACTORIAL\_02 (N in number)**

**RETURN NUMBER IS**

**RES NUMBER:=1;**

**TEMP NUMBER:=N;**

**BEGIN**

**WHILE TEMP>0 LOOP**

**RES:=TEMP\*RES;**

**TEMP:=TEMP-1;**

**END LOOP;**

**RETURN RES;**

**END;**

**/**

**DECLARE**

**N NUMBER;**

**BEGIN**

**N := &N;**

**DBMS\_OUTPUT.PUT\_LINE('FACTORIAL OF '|| N);**

**DBMS\_OUTPUT.PUT\_LINE(FACTORIAL\_02(N));**

**END;**

**/**

**Output:**

SQL> @ E:\HammadDBMS\PLSQL\_FUNCTION\func\_1\_CALL.SQL

Enter value for n: 4

old 5: N := &N;

new 5: N := 4;

ENTER A NUMBER TO FIND FACTORIAL

24

PL/SQL procedure successfully completed.

SQL> /

Enter value for n: 5

old 5: N := &N;

new 5: N := 5;

ENTER A NUMBER TO FIND FACTORIAL

120

2. Write a PL/SQL block to find the factorial of a number using recursive function.

**Code:**

**DECLARE**

**N NUMBER;**

**RES NUMBER;**

**CREATE OR REPLACE FUNCTION FACT\_R\_02(X NUMBER)**

**RETURN NUMBER**

**IS**

**F NUMBER;**

**BEGIN**

**IF X=0 THEN**

**F := 1;**

**ELSE**

**F := X \* FACT\_R\_02(X-1);**

**END IF;**

**RETURN F;**

**END FACT\_R\_02;**

**BEGIN**

**N:= &N;**

**RES := FACT\_R\_02(N);**

**dbms\_output.put\_line(' Factorial '|| N || ' is ' || RES);**

**END;**

**/**

**Output:**

SQL> @ E:\HammadDBMS\PLSQL\_FUNCTION\func\_2.SQL

Enter value for n: 4

old 18: N:= &N;

new 18: N:= 4;

Factorial 4 is 24

PL/SQL procedure successfully completed.

SQL> /

Enter value for n: 5

old 18: N:= &N;

new 18: N:= 5;

Factorial 5 is 120

PL/SQL procedure successfully completed.

3. Write a PL/SQL block to demonstrate Function Overloading.

**Code:**

**SET SERVEROUTPUT ON;**

**DECLARE**

**N1 NUMBER;**

**N2 NUMBER;**

**N3 NUMBER;**

**RES1 NUMBER;**

**FUNCTION ADD\_02(X NUMBER, Y NUMBER)**

**RETURN NUMBER**

**IS**

**R NUMBER;**

**BEGIN**

**R := X + Y;**

**RETURN R;**

**END ADD\_02;**

**FUNCTION ADD\_02(X NUMBER, Y NUMBER,Z NUMBER)**

**RETURN NUMBER**

**IS**

**R NUMBER;**

**BEGIN**

**R := X + Y + Z;**

**RETURN R;**

**END ADD\_02;**

**BEGIN**

**N1 := &N;**

**N2 := &N;**

**N3 := &N;**

**RES1 := ADD\_02(N1,N2);**

**dbms\_output.put\_line(' ADDITION OF FIRST TWO NUMBERS ' || RES1);**

**RES1 := ADD\_02(N1,N2,N3);**

**dbms\_output.put\_line(' ADDITION OF THREE NUMBERS ' || RES1);**

**END;**

**/**

**Output:**

SQL> @ E:\HammadDBMS\PLSQL\_FUNCTION\FUNC\_3.SQL

Enter value for n: 3

old 25: N1 := &N;

new 25: N1 := 3;

Enter value for n: 4

old 26: N2 := &N;

new 26: N2 := 4;

Enter value for n: 5

old 27: N3 := &N;

new 27: N3 := 5;

ADDITION OF FIRST TWO NUMBERS 7

ADDITION OF THREE NUMBERS 12

PL/SQL procedure successfully completed.

4. Write a PL/SQL block to display name of depositor having highest bank amount using function.

**Code:**

**SET SERVEROUTPUT ON;**

**DECLARE**

**FUNCTION HIGH\_02**

**return VARCHAR as**

**NAME VARCHAR(20);**

**CURSOR c1 IS SELECT \* FROM DEPOSIT\_02;**

**r c1%ROWTYPE;**

**MAX DEPOSIT\_02.AMOUNT%TYPE:=0;**

**BEGIN**

**MAX:=0;**

**FOR r IN c1 LOOP**

**MAX:=r.amount;**

**NAME:=r.cname;**

**END LOOP;**

**RETURN NAME;**

**END HIGH\_02;**

**BEGIN**

**dbms\_output.put\_line(' NAME OF THE HIGHEST DEPOSITOR IS ' || HIGH\_02());**

**END;**

**/**

**Output:**

SQL> @ E:\HammadDBMS\PLSQL\_FUNCTION\FUNC\_4.SQL

Function created.

**NAME OF THE HIGHEST DEPOSITOR IS NAREN**

PL/SQL procedure successfully completed.

5. Write a PL/SQL block to display number of depositors using function.

**Code:**

**SET SERVEROUTPUT ON;**

**DECLARE**

**CNT NUMBER;**

**FUNCTION COUNT\_02**

**return VARCHAR as**

**NAME VARCHAR(20);**

**CURSOR c1 IS SELECT \* FROM DEPOSIT\_02;**

**r c1%ROWTYPE;**

**BEGIN**

**CNT:=0;**

**OPEN c1;**

**FETCH c1 into r;**

**WHILE(c1%found)LOOP**

**CNT:=CNT+1;**

**FETCH c1 INTO r;**

**END LOOP;**

**RETURN CNT;**

**END COUNT\_02;**

**BEGIN**

**dbms\_output.put\_line(' NUMBER OF DEPOSITORS : ' || COUNT\_02());**

**END;**

**/**

**Output:**

SQL> @ E:\HammadDBMS\PLSQL\_FUNCTION\FUNC\_5.SQL

Function created.

**NUMBER OF DEPOSITORS : 9**

PL/SQL procedure successfully completed.

6. Write a PL/SQL block to display number of depositors whose name starts with A using function.

**Code:**

**SET SERVEROUTPUT ON;**

**DECLARE**

**CNT NUMBER;**

**FUNCTION COUNT2\_02**

**return VARCHAR as**

**NAME VARCHAR(20);**

**CURSOR c1 IS SELECT \* FROM DEPOSIT\_02 WHERE CNAME LIKE 'A%';**

**r c1%ROWTYPE;**

**BEGIN**

**CNT:=0;**

**OPEN c1;**

**FETCH c1 into r;**

**WHILE(c1%found)LOOP**

**CNT:=CNT+1;**

**FETCH c1 INTO r;**

**END LOOP;**

**RETURN CNT;**

**END COUNT2 \_02;**

**BEGIN**

**dbms\_output.put\_line(' NUMBER OF DEPOSITOR WHOSE NAME STARTS WITH A : ' || COUNT2\_02());**

**END;**

**/**

**Output:**

SQL> @ E:\HammadDBMS\PLSQL\_FUNCTION\FUNC\_6.SQL

Function created.

**NUMBER OF DEPOSITOR WHOSE NAME STARTS WITH A : 1**

PL/SQL procedure successfully completed.

7. Write a PL/SQL block to display branch name with fifth highest amount in deposit using function.

**Code:**

**SET SERVEROUTPUT ON;**

**DECLARE**

**FUNCTION FIFTH\_02**

**RETURN VARCHAR AS**

**NAME VARCHAR(20);**

**CURSOR C1 IS SELECT BNAME BM FROM DEPOSIT\_02 ORDER BY AMOUNT DESC;**

**R C1%ROWTYPE;**

**BEGIN**

**FOR R IN C1 LOOP**

**IF(C1%rowcount = 5)THEN**

**NAME := R.BM;**

**END IF;**

**END LOOP;**

**RETURN NAME;**

**END FIFTH\_02;**

**BEGIN**

**dbms\_output.put\_line('NAME OF THE BRANCH WITH FIFTH HIGHEST AMOUNT : ' || FIFTH\_02());**

**END;**

**/**

**Output:**

SQL> @E:\HammadDBMS\PLSQL\_FUNCTION\FUNC\_7.SQL

NAME OF THE BRANCH WITH FIFTH HIGHEST AMOUNT : M.G.ROAD

PL/SQL procedure successfully completed.

SQL> SPOOL OFF;

**PROCEDURES:**

1. Write a PL/SQL block to display depositor name and date whose opening account is after 12/3/1998 using procedure.

**Code:**

**create or replace procedure opening\_after\_date**

**is**

**cursor c is SELECT cname, adate FROM deposit\_02 WHERE adate > '12-MAR-98';**

**is\_found\_rec boolean := false;**

**not\_found exception;**

**BEGIN**

**for i in c loop**

**is\_found\_rec := true;**

**dbms\_output.put\_line(i.cname || ' ' || i.adate);**

**end loop;**

**if not is\_found\_rec THEN**

**raise not\_found;**

**end if;**

**exception**

**when not\_found then**

**dbms\_output.put\_line('No depositors joined after 12/3/1998.');**

**end opening\_after\_date;**

**/**

**DECLARE**

**BEGIN**

**opening\_after\_date();**

**end;**

**/**

**Output:**

SQL> E:\HammadDBMS\PLSQL\_PROCEDURES\PRO\_1.SQL

Procedure created.

No depositors joined after 12/3/1998.

PL/SQL procedure successfully completed.

2. Write a PL/SQL block to display depositor details of a specific name using procedure.

**Code:**

**SERVEROUTPUT ON;**

**DECLARE**

**CUST VARCHAR2(20);**

**PROCEDURE PRO\_2(CUSTNAME IN VARCHAR2) IS**

**CURSOR C1 IS SELECT \* FROM DEPOSIT\_02 WHERE CNAME = CUSTNAME;**

**BEGIN**

**FOR R IN C1 LOOP**

**DBMS\_OUTPUT.PUT\_LINE('ACCOUNT NUMBER : '|| R.AC\_NO);**

**DBMS\_OUTPUT.PUT\_LINE('NAME : '|| R.CNAME);**

**DBMS\_OUTPUT.PUT\_LINE('BRANCH : '|| R.BNAME);**

**DBMS\_OUTPUT.PUT\_LINE('AMOUNT : '|| R.AMOUNT);**

**DBMS\_OUTPUT.PUT\_LINE('CREATE DATE : '|| R.ADATE);**

**END LOOP;**

**END PRO\_2;**

**BEGIN**

**CUST := '&CUST';**

**PRO\_2(CUST);**

**END;**

**/**

**Output:**

SQL> @E:\HammadDBMS\PLSQL\_PROCEDURES\PRO\_2.SQL

Enter value for cust: ANIL

old 16: CUST := '&CUST';

new 16: CUST := 'ANIL';

ACCOUNT NUMBER : 100

NAME : ANIL

BRANCH : VRCE

AMOUNT : 1120

CREATE DATE : 01-MAR-95

PL/SQL procedure successfully completed.

SQL> SPOOL OFF;

3. Write a PL/SQL block to update (replace) all the names of customers with first character capital and others in lower case using procedure.

**Code:**

**create or replace procedure capitalize\_first\_letter**

**is**

**BEGIN**

**UPDATE customer\_02 SET cname = INITCAP(cname);**

**UPDATE deposit\_02 SET cname = INITCAP(cname);**

**UPDATE borrow\_02 SET cname = INITCAP(cname);**

**end capitalize\_first\_letter;**

**/**

**DECLARE**

**BEGIN**

**capitalize\_first\_letter();**

**dbms\_output.put\_line('First letters of all names are not capital, and the rest are small.');**

**end;**

**/**

**Output:**

SQL> ALTER TABLE deposit\_02 DISABLE CONSTRAINT FK\_Customer;

Table altered.

SQL> ALTER TABLE borrow\_02 DISABLE CONSTRAINT FK\_Customer1;

Table altered.

SQL> select \* from customer\_02;

CNAME CITY

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

ANIL KOLKATA

SUNIL DELHI

MEHUL BARODA

MANDAR PATNA

MADHURI NAGPUR

PRAMOD NAGPUR

SANDIP SURAT

SHIVANI MUMBAI

KRANTI MUMBAI

NAREN MUMBAI

10 rows selected.

SQL> @E:\HammadDBMS\PLSQL\_PROCEDURES\PRO\_3.SQL

Procedure created.

First letters of all names are not capital, and the rest are small.

PL/SQL procedure successfully completed.

SQL> select \* from customer\_02;

CNAME CITY

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

Anil KOLKATA

Sunil DELHI

Mehul BARODA

Mandar PATNA

Madhuri NAGPUR

Pramod NAGPUR

Sandip SURAT

Shivani MUMBAI

Kranti MUMBAI

Naren MUMBAI

10 rows selected.

4. Write a PL/SQL block to display amount in the format 99,999,99 using procedure.

**Code:**

**create or replace procedure change\_number\_format**

**is**

**cursor c is SELECT amount FROM deposit\_02;**

**BEGIN**

**for i in c loop**

**dbms\_output.put\_line(to\_char(i.amount, '99,999,99'));**

**end loop;**

**end change\_number\_format;**

**/**

**DECLARE**

**BEGIN**

**change\_number\_format();**

**end;**

**/**

**Output:**

SQL> @E:\HammadDBMS\PLSQL\_PROCEDURES\PRO\_4.SQL

Procedure created.

10,00

50,00

35,00

12,00

30,00

20,00

10,00

50,00

70,00

PL/SQL procedure successfully completed.

5. Write a PL/SQL block to display average amount of depositors using procedure.

**Code:**

**create or replace procedure display\_average\_amount(average out number)**

**is**

**BEGIN**

**SELECT AVG(amount) INTO average FROM deposit\_02;**

**end display\_average\_amount;**

**/**

**DECLARE**

**average number(8,2);**

**BEGIN**

**display\_average\_amount(average);**

**dbms\_output.put\_line('The average amount is ' || average);**

**end;**

**/**

**Output:**

SQL> @E:\HammadDBMS\PLSQL\_PROCEDURES\PRO\_5.SQL

Procedure created.

The average amount is 3188.89

PL/SQL procedure successfully completed.

**PACKAGES:**

1. Create a package which consists of a function of addition of two numbers and a procedure for multiplication of two numbers.

**Code:**

**create or replace package arithmetic\_02 as**

**function addition(a in number, b in number) return number;**

**procedure multiplication(a in number, b in number, c out number);**

**end arithmetic\_02;**

**/**

**create or replace package body arithmetic\_02 as**

**function addition\_02(a in number, b in number)**

**return number is**

**begin**

**return a + b;**

**end addition\_02;**

**procedure multiplication\_02(a in number, b in number, c out number)**

**is**

**begin**

**c := a \* b;**

**end multiplication\_02;**

**end arithmetic\_02;**

**/**

**declare**

**a number;**

**b number;**

**c number;**

**begin**

**a := &a;**

**b := &b;**

**dbms\_output.put\_line('Sum of the numbers: ' || arithmetic\_02.addition\_02(a, b));**

**arithmetic\_02.multiplication\_02(a, b, c);**

**dbms\_output.put\_line('Product of the numbers: ' || c);**

**end;**

**/**

**Output:**

SQL> @E:\HammadDBMS\PLSQL\_PACKAGES\PACK\_1.SQL

Package created.

Package body created.

Enter value for a: 3

old 6: a := &a;

new 6: a := 3;

Enter value for b: 4

old 7: b := &b;

new 7: b := 4;

Sum of the numbers: 7

Product of the numbers: 12

PL/SQL procedure successfully completed.

2. Create a package which consists of a function of factorial of a number and a procedure of factorial of a two number.

**Code:**

**create or replace package factorial\_02 AS**

**function factorial\_one(a in number) return number;**

**procedure factorial\_two(a in number, b in number, fact1 out number, fact2 out number);**

**end factorial\_02;**

**/**

**create or replace package body factorial\_02 as**

**function factorial\_one(a in number)**

**return number is**

**fact number := 1;**

**begin**

**for i in 1 .. a loop**

**fact := fact \* i;**

**end loop;**

**return fact;**

**end factorial\_one;**

**procedure factorial\_two(a in number, b in number, fact1 out number, fact2 out number)**

**is**

**begin**

**fact1 := 1;**

**fact2 := 1;**

**for i in 1 .. a loop**

**fact1 := fact1 \* i;**

**end loop;**

**for i in 1 .. b loop**

**fact2 := fact2 \* i;**

**end loop;**

**end factorial\_two;**

**end factorial\_02;**

**/**

**DECLARE**

**a number;**

**b number;**

**c number;**

**fact1 number;**

**fact2 number;**

**BEGIN**

**a := &a;**

**b := &b;**

**c := &c;**

**dbms\_output.put\_line('Factorial of ' || a || ': ' || factorial\_02.factorial\_one(a));**

**factorial\_02.factorial\_two(b, c, fact1, fact2);**

**dbms\_output.put\_line('Factorial of ' || b || ': ' || fact1);**

**dbms\_output.put\_line('Factorial of ' || c || ': ' || fact2);**

**end;**

**/**

**Output:**

SQL> @E:\HammadDBMS\PLSQL\_PACKAGES\PACK\_2.SQL

Package created.

Package body created.

Enter value for a: 5

old 8: a := &a;

new 8: a := 5;

Enter value for b: 4

old 9: b := &b;

new 9: b := 4;

Enter value for c: 6

old 10: c := &c;

new 10: c := 6;

Factorial of 5: 120

Factorial of 4: 24

Factorial of 6: 720

PL/SQL procedure successfully completed.

3. Create a package which consists of function to display name of depositor having highest bank amount and a procedure to display depositor name and date whose opening account is after 12/3/1998 using procedure.

**Code:**

**create or replace package highest\_02 as**

**function highest\_amt return varchar2;**

**procedure opening\_after\_date;**

**end highest\_02;**

**/**

**create or replace package body highest\_02 as**

**function highest\_amt**

**return varchar2 is**

**n varchar2(20);**

**BEGIN**

**SELECT cname into n FROM deposit\_02 WHERE amount in(SELECT MAX(amount) FROM deposit\_02);**

**return n;**

**end highest\_amt;**

**procedure opening\_after\_date**

**is**

**cursor c is SELECT cname, adate FROM deposit\_02 WHERE adate > '12-MAR-98';**

**is\_found\_rec boolean := false;**

**not\_found exception;**

**BEGIN**

**for i in c loop**

**is\_found\_rec := true;**

**dbms\_output.put\_line(i.cname || ' ' || i.adate);**

**end loop;**

**if not is\_found\_rec THEN**

**raise not\_found;**

**end if;**

**exception**

**when not\_found then**

**dbms\_output.put\_line('No depositors joined after 12/3/1998.');**

**end opening\_after\_date;**

**end highest\_02;**

**/**

**DECLARE**

**name varchar2(20);**

**BEGIN**

**name:= highest\_02.highest\_amt();**

**dbms\_output.put\_line(name || ' has the highest amount deposited in the bank.');**

**highest\_02.opening\_after\_date();**

**end;**

**/**

**Output:**

SQL> @E:\HammadDBMS\PLSQL\_PACKAGES\PACK\_3.SQL

Package created.

Package body created.

NAREN has the highest amount deposited in the bank.

No depositors joined after 12/3/1998.

PL/SQL procedure successfully completed.

4. Create a package which consists of a procedure to display amount in the format 99,999,99 and a function to display average amount of depositors.

**Code:**

**create or replace package average\_02 AS**

**procedure change\_number\_format;**

**function display\_average\_amount return number;**

**end average\_02;**

**/**

**create or replace package body average\_02 AS**

**procedure change\_number\_format**

**is**

**cursor c is SELECT amount FROM deposit\_02;**

**BEGIN**

**for i in c loop**

**dbms\_output.put\_line(to\_char(i.amount, '99,999,99'));**

**end loop;**

**end change\_number\_format;**

**function display\_average\_amount**

**return number**

**is**

**average number(8, 2);**

**BEGIN**

**SELECT AVG(amount) INTO average FROM deposit\_02;**

**return average;**

**end display\_average\_amount;**

**end average\_02;**

**/**

**DECLARE**

**BEGIN**

**dbms\_output.put\_line('Average is ' || average\_02.display\_average\_amount());**

**dbms\_output.put\_line('Amounts with changed format:');**

**average\_02.change\_number\_format();**

**end;**

**/**

**Output:**

SQL> @E:\HammadDBMS\PLSQL\_PACKAGES\PACK\_5.SQL

Package created.

Package body created.

Average is 3188.89

Amounts with changed format:

10,00

50,00

35,00

12,00

30,00

20,00

10,00

50,00

70,00

PL/SQL procedure successfully completed.