RELATIONAL DATABASE MANAGEMENT SYSTEM - II

Name: BIND SUNNY SANTOSH
Roll No: CS22009

ATMANAND SARASWATI SCIENCE COLLEGE
KAPODRA, VARACHHA ROAD, SURAT – 395006.

1. Create simple PL/SQL Block which prints 'Welcome to PL/SQL World'.

INPUT:

```
BEGIN
 DBMS_OUTPUT.PUT_LINE('Welcome TO PL/SQL  WORLD');
END;
```

OUTPUT:

```
Welcome TO PL/SQL  WORLD

Statement processed.
```

2. Create Sequence which generates number from 1 upto 999 and don't repeat it.

INPUT:

```
CREATE SEQUENCE sqn
START WITH 1
INCREMENT BY 1
MINVALUE 1
MAXVALUE 999
NOCYCLE;

SELECT COUNT1.NEXTVAL FROM DUAL;
```

OUTPUT:

```
Sequence created.
```

3. Create custom data type Address_TY. (HouseNo, Address, Road, City, Pincode)

INPUT:

```
CREATE TYPE ADDRESS_TY AS OBJECT
 (
HouseNo VARCHAR(4),
 Address VARCHAR(50),
 Road VARCHAR(30),
 City VARCHAR(10),
 Pincode NUMBER(6)
);
```

OUTPUT:

```
Type created.
```

4. Create PL/SQL Block which prints 'Hello World' 5 times.

INPUT:

```
DECLARE
  i NUMBER;
BEGIN
 for i IN 1..5
 LOOP
 DBMS_OUTPUT.PUT_LINE('HELLO WORLD');
 END LOOP;
END;
```

OUTPUT:

```
HELLO WORLD
HELLO WORLD
HELLO WORLD
HELLO WORLD
HELLO WORLD

Statement processed.
```

5. Create PL/SQL Block which takes message and number form user and then prints that message n times on the screen.

INPUT:

```
DECLARE
 MESSAGE VARCHAR(50);
 N NUMBER;
BEGIN
 MESSAGE:=:MESSAGE;
 N:=:N;
 FOR I IN 1..N
 LOOP
 DBMS_OUTPUT.PUT_LINE(MESSAGE);
 END LOOP;
END;
```

OUTPUT:

```
RDBMS PART-2
RDBMS PART-2
RDBMS PART-2
RDBMS PART-2
RDBMS PART-2
RDBMS PART-2

Statement processed.
```

6. Create PL/SQL Block which takes radius of circle from user and then calculates the area of circle and then print it.

INPUT:

```
DECLARE
```

```
 radius NUMBER;
 AOC NUMBER;
 PI NUMBER;
BEGIN
 PI:= 3.14;
 radius:=:radius;
 AOC:=PI*radius*radius;
 DBMS_OUTPUT.PUT_LINE('Area Of Circle is '||AOC);
END;
```

OUTPUT:

```
Area Of Circle is 12.56

Statement processed.
```

7. Create PL/SQL block which takes Principle amount, Interest Rate and time from user then calculate simple interest (I=(P*R*N)/100) and print it.

INPUT:

```
DECLARE
 PRINCIPAL NUMBER;
 RATE NUMBER;
 TIME NUMBER;
 INTEREST NUMBER;
BEGIN
 PRINCIPAL:=:PRINCIPAL;
 RATE:=:RATE;
 TIME:=:TIME;
 INTEREST:=(PRINCIPAL*RATE*TIME)/100;
 DBMS_OUTPUT.PUT_LINE('INTEREST PER MONTH IS '||INTEREST ||' RUPEES');
END;
```

OUTPUT:

```
INTEREST PER MONTH IS 100 RUPEES
```

```
Statement processed.
```

8. Create PL/SQL block which takes number from user and check whether it is positive or negative.

INPUT:

```
DECLARE
 N NUMBER;
BEGIN
 N:=:N;
 IF(N>0)
THEN
DBMS_OUTPUT.PUT_LINE(N||' IS A POSITIVE NUMBER.');
ELSIF(N<0)
THEN
DBMS_OUTPUT.PUT_LINE(N||' IS A NEGATIVE NUMER.');
ELSE
 DBMS_OUTPUT.PUT_LINE(N||',N IS A ZERO.');
END IF;
END;
```

OUTPUT:

```
5 IS A POSITIVE NUMBER.

Statement processed.
```

9. Create PL/SQL block for printing Fibonacci series. (0 1 1 2 3 5 8 13 ……. Summation of previous two numbers)

INPUT:

```
DECLARE
 FIRST NUMBER:=0;
 SECOND NUMBER:=1;
 TEMP NUMBER;
 I NUMBER;
```

```
 N NUMBER;
BEGIN
 N:=:N;
 DBMS_OUTPUT.PUT_LINE('FIBONACCI SERIES');
 DBMS_OUTPUT.PUT_LINE(FIRST);
 DBMS_OUTPUT.PUT_LINE(SECOND);
 FOR L IN 2..N
 LOOP
 TEMP:=FIRST+SECOND;
 FIRST:=SECOND;
 SECOND:=TEMP;
 DBMS_OUTPUT.PUT_LINE(TEMP);
END LOOP;
END;
```

OUTPUT:

```
FIBONACCI SERIES
0
1
1
2
3
5

Statement processed.
```

10. Create PL/SQL block to display Multiplication table.

INPUT:

```
DECLARE
 NUM1 NUMBER:=10;
BEGIN
 DBMS_OUTPUT.PUT_LINE('MULTIPLICATION TABLE OF 10 ');
 FOR NUM2 IN 1..10
 LOOP
 DBMS_OUTPUT.PUT_LINE(NUM1||'*'||NUM2||'='||NUM1*NUM2);
END LOOP;
END;
```

OUTPUT:

```
MULTIPLICATION TABLE OF 10
10*1=10
10*2=20
10*3=30
10*4=40
10*5=50
10*6=60
10*7=70
10*8=80
10*9=90
10*10=100

Statement processed.
```

11. Create PL/SQL block to display reverse a Number. (Take input from user)

INPUT:

```
DECLARE
 NUM NUMBER;
 REV NUMBER;
BEGIN
 rev:=0;
 NUM:=:NUM;
 DBMS_OUTPUT.PUT_LINE('THE NUMBER IS '||NUM);
 WHILE NUM>0 LOOP
 rev:=(rev*10) + mod(num,10);
 NUM:=floor(NUM/10);
 END LOOP;
 DBMS_OUTPUT.PUT_LINE('Reverse of the number is: ' || rev);
END;
```

OUTPUT:

```
THE NUMBER IS 456
Reverse of the number is: 654

Statement processed.
```

12. Create PL/SQL block to display summation of n numbers.

INPUT:

```
DECLARE
 ADD NUMBER:=0;
 N NUMBER;
BEGIN
 DBMS_OUTPUT.PUT_LINE('YOU WANT THE SUMMATION OF UPTO HOW MANY NUMBER
');
 N:=:N;
 FOR I IN 1..N
 LOOP
 ADD:=ADD+I*(I+1)/2;
 END LOOP;
 DBMS_OUTPUT.PUT_LINE('SUMMATION OF '||N||' IS '||ADD);
END;
```

OUTPUT:

```
YOU WANT THE SUMMATION OF UPTO HOW MANY NUMBER
SUMMATION OF 5 IS 35

Statement processed.
```

13. Create PL/SQL block to Display Factorial of Number. (Take input from user)

INPUT:

```
DECLARE
 FACTORIAL NUMBER:=1;
 N NUMBER;
BEGIN
 N:=:N;
 DBMS_OUTPUT.PUT_LINE('N IS '||N);
```

```
WHILE N>0
LOOP
FACTORIAL:=FACTORIAL*N;
N:=N-1;
END LOOP;
DBMS_OUTPUT.PUT_LINE('FACTORIAL IS '||FACTORIAL);
END;
```

OUTPUT:

```
N IS 5
FACTORIAL OF IS 120

Statement processed.
```

14. Create PL/SQL block to calculate the area of a circle for a value of radius from 5 to 15. Store the radius and the corresponding values of calculated area in an empty table named AreaMst consist of two columns radius and area. (Hint: use power function)

INPUT:

```
CREATE TABLE AREAMST
(RADIUS NUMBER PRIMARY KEY,
AREA NUMBER(14,2) NOT NULL);

DECLARE
 R NUMBER;
 PI NUMBER;
 AREA NUMBER;
BEGIN
 PI:= 3.14;
 R:=5;
 FOR R IN 9..15
 LOOP
 AREA:=PI*POWER(R,2);
 INSERT INTO AREAMST VALUES(R,AREA);
 END LOOP;
```

```
END;

SELECT * FROM AREAMST;
```

OUTPUT:

```
Table created.
```

RADIUS	AREA
9	254.34
10	314
11	379.94
12	452.16
13	530.66
14	615.44
15	706.5

7 rows returned in 0.00 seconds

15. Create table EmployeeMst and use ADDRESS_TY as address column datatype. (EID, FNAME, MNAME, LNAME, Gender, Address, DOB, DOJ, Salary, Email, Contact, Designation)

INPUT:

```
CREATE TABLE EmployeeMst
(
  EID NUMBER PRIMARY KEY,
  FNAME VARCHAR(30),
  MNAME VARCHAR(30),
  LNAME VARCHAR(30),
  Gender VARCHAR(6),
  Address ADDRESS_TY,
  DOB DATE,
  DOJ DATE,
```

```
 Salary NUMBER,,
 Email VARCHAR(50),
 Contact VARCHAR(15),
 Designation VARCHAR(50)
);
```

OUTPUT:

```
Table created.
```

16. Insert 15 record in EmployeeMst. (Make use of Sequence and take a look at how to insert data into user defined datatype)

INPUT:

```
INSERT INTO EmployeeMst (EID, FName, MName,LName, Gender, Address, DOB, DOJ, Salary, Email,Contact,Designation)
VALUES (sqn.NextVal, 'Pallavi', 'Jagjivanbhai','Thummar','Female',ADDRESS_TY('304','Indralok Res.','p.t.road','Surat', 364101), '20-dec-2001', '20-apr-2019', 1000000,'pallavi@gmail.com','9012345678','CEO');

INSERT INTO EmployeeMst (EID, FName, MName,LName, Gender, Address, DOB, DOJ, Salary, Email,Contact,Designation)
VALUES (sqn.NextVal, 'Palak','Vipulbhai','Patel','Female',ADDRESS_TY('103','Paredise','p.t.road','Surat',364101),' 01-jan-2000', '02-sep-2019',10000, 'palak@gmail.com','9012345678','Manager');

INSERT INTO EmployeeMst (EID, FName, MName,LName, Gender, Address, DOB, DOJ, Salary, Email,Contact,Designation)
VALUES (sqn.NextVal, 'Arun', 'Miteshbhai', 'Rabdiya','Male',ADDRESS_TY('404','RiverView','p.t.road','Surat', 364101), '09-jan-2000', '02-mar-2018', 10000,'arun@gmail.com','9012348968','Cleark');


INSERT INTO EmployeeMst (EID, FName, MName,LName, Gender, Address, DOB, DOJ, Salary, Email,Contact,Designation)
VALUES (sqn.NextVal, 'Shruti', 'Vipulbhai','Patel','Female',ADDRESS_TY('101','Rivera','p.t.road','Surat',364101), '01-dec-2001', '05-jul-2019', 10000,'shruti@gmail.com','9014561678','Peon');
```

```
INSERT INTO EmployeeMst (EID, FName, MName,LName, Gender, Address, DOB, DOJ, Salary, Email,Contact,Designation)
VALUES (sqn.NextVal, 'Ritu', 'Riteshbhai','Gondaliya','Female',ADDRESS_TY('802','Rajpalace','p.t.road','Vadodra',364101), '26-feb-2001', '22-mar-2015',45000, 'ritu@gmail.com','9012385278','Cleark');

INSERT INTO EmployeeMst (EID, FName, MName,LName, Gender, Address, DOB, DOJ, Salary, Email,Contact,Designation)
VALUES (sqn.NextVal, 'Karina', 'Kiritbhai','Kapoor','Female',ADDRESS_TY('403','Rajhans','p.t.road','Bharuch', 364101), '01-apr-2003','02-mar-2018', 50000,'karina@gmail.com','9012345852','Manager');

INSERT INTO EmployeeMst (EID, FName, MName,LName, Gender, Address, DOB, DOJ, Salary, Email,Contact,Designation)
VALUES (sqn.NextVal, 'Pritesh' , 'Rafikbhai' , 'Mahmand','Male', ADDRESS_TY('1','Hello' , 'p.t.road','Arebia',364101), '12-jan-1998' , '02-mar-2019' , 10000 , 'pritesh@gmail.com','9612345678','Peon' );

INSERT INTO EmployeeMst (EID, FName, MName,LName, Gender, Address, DOB, DOJ, Salary, Email,Contact,Designation)
VALUES (sqn.NextVal, 'Darshan', 'Maheshbhai','Patel','Male', ADDRESS_TY('204','ABC','p.t.road','Bhavnagar', 364101),'01-dec-1998','02-jul-2014', 10000,'darshan@gmail.com','9634125678','Manager');

INSERT INTO EmployeeMst (EID, FName, MName,LName, Gender, Address, DOB, DOJ, Salary, Email,Contact,Designation)
VALUES (sqn.NextVal, 'Soham', 'Sahilbhai', 'Vyas','Male',ADDRESS_TY('103','Paredise','p.t.road','Rajkot', 364101), '26-feb-2001', '22-sep-2019', 35000, 'soham@gmail.com','8522345678','Peon');

INSERT INTO EmployeeMst (EID, FName, MName,LName, Gender, Address, DOB, DOJ, Salary, Email,Contact,Designation)VALUES (sqn.NextVal, 'Pinank', 'Parimalbhai','Thummar','Male',ADDRESS_TY('101','XYZ','p.t.road','Amreli', 364101), '01-jan-2000', '02-aug-2018', 10000, 'p123@gmail.com','9612632678','Security');
```

OUTPUT:

EID	FNAME	MNAME	LNAME	GENDER	DOB	DOJ	SALARY	EMAIL	CONTACT	DESIGNATION
1	Pallavi	Jagjivanbhai	Thummar	Female	20-DEC-01	20-APR-19	1000000	pallavi@gmail.com	9012345678	CEO
2	Palak	Vipulbhai	Patel	Female	01-JAN-00	02-SEP-19	10000	palak@gmail.com	9012345678	Manager
3	Arun	Miteshbhai	Rabdiya	Male	09-JAN-00	02-MAR-18	10000	arun@gmail.com	9012348968	Cleark
4	Shruti	Vipulbhai	Patel	Female	01-DEC-01	05-JUL-19	10000	shruti@gmail.com	9014561678	Peon
5	Ritu	Riteshbhai	Gondaliya	Female	26-FEB-01	22-MAR-15	45000	ritu@gmail.com	9012385278	Cleark
6	Karina	Kiritbhai	Kapoor	Female	01-APR-03	02-MAR-18	50000	karina@gmail.com	9012345852	Manager
7	Pritesh	Rafikbhai	Mahmand	Male	12-JAN-98	02-MAR-19	10000	pritesh@gmail.com	9612345678	Peon
8	Darshan	Maheshbhai	Patel	Male	01-DEC-98	02-JUL-14	10000	darshan@gmail.com	9634125678	Manager
9	Soham	Sahilbhai	Vyas	Male	26-FEB-01	22-SEP-19	35000	soham@gmail.com	8522345678	Peon
10	Pinank	Parimalbhai	Thummar	Male	01-JAN-00	02-AUG-18	10000	p123@gmail.com	9612632678	Security

10 rows returned in 0.00 seconds CSV Export

17. Create PL/SQL Block to make use of goto Keyword. (Create your own Example)

INPUT:

```
DECLARE
  v_counter NUMBER := 1;
BEGIN
  <<start_loop>>
  LOOP
    DBMS_OUTPUT.PUT_LINE('Counter: ' || v_counter);
    v_counter := v_counter + 1;
    IF v_counter = 5 THEN
      GOTO exit_loop;
    END IF;
  END LOOP;
  <<exit_loop>>
  DBMS_OUTPUT.PUT_LINE('Exiting the loop.');
END;
```

OUTPUT:

```
Counter: 1
Counter: 2
Counter: 3
Counter: 4
Exiting the loop.

Statement processed.
```

18. Update Salary of Employee ID 1 using PL/SQL Block and print proper message whether the record is updated or not. (Hint: Use Implicit Cursor Attributes as well as fire commit before the END)

INPUT:

```
DECLARE
  v_employee_id NUMBER := 1;
  v_new_salary NUMBER := 60000;
BEGIN
  UPDATE EmployeeMst
  SET Salary = v_new_salary
  WHERE EID = v_employee_id;
  IF SQL%FOUND THEN
    DBMS_OUTPUT.PUT_LINE('Salary updated successfully for Employee ID ' ||
v_employee_id);
    COMMIT; -- Commit the transaction
  ELSIF SQL%NOTFOUND THEN
    DBMS_OUTPUT.PUT_LINE('Employee ID ' || v_employee_id || ' not found. No update
performed.');
  ELSE
    DBMS_OUTPUT.PUT_LINE('Error during the update operation.');
  END IF;
END;
```

OUTPUT:

EID	FNAME	MNAME	LNAME	GENDER	DOB	DOJ	SALARY	EMAIL	CONTACT	DESIGNATION
1	Pallavi	Jagjivanbhai	Thummar	Female	20-DEC-01	20-APR-19	60000	pallavi@gmail.com	9012345678	CEO

19. Update Salary of all Peon using PL/SQL Block and print how many records are updated. (Hint: Use Implicit Cursor Attributes as well as fire commit before the END)

INPUT:

```
DECLARE
  v_updated_count NUMBER := 0;
BEGIN
```

```
  UPDATE EmployeeMst
  SET Salary = Salary * 1.1
  WHERE Designation = 'Peon';
  v_updated_count := SQL%ROWCOUNT;

  IF v_updated_count > 0 THEN
    DBMS_OUTPUT.PUT_LINE(v_updated_count || ' records updated successfully.');
    COMMIT;
  ELSE
    DBMS_OUTPUT.PUT_LINE('No records updated. No changes to commit.');
  END IF;
END;
```

OUTPUT:

EID	FNAME	MNAME	LNAME	GENDER	DOB	DOJ	SALARY	EMAIL	CONTACT	DESIGNATION
1	Pallavi	Jagjivanbhai	Thummar	Female	20-DEC-01	20-APR-19	60000	pallavi@gmail.com	9012345678	CEO
2	Palak	Vipulbhai	Patel	Female	01-JAN-00	02-SEP-19	10000	palak@gmail.com	9012345678	Manager
3	Arun	Miteshbhai	Rabdiya	Male	09-JAN-00	02-MAR-18	10000	arun@gmail.com	9012348968	Cleark
4	Shruti	Vipulbhai	Patel	Female	01-DEC-01	05-JUL-19	11000	shruti@gmail.com	9014561678	Peon
5	Ritu	Riteshbhai	Gondaliya	Female	26-FEB-01	22-MAR-15	45000	ritu@gmail.com	9012385278	Cleark
6	Karina	Kiritbhai	Kapoor	Female	01-APR-03	02-MAR-18	50000	karina@gmail.com	9012345852	Manager
7	Pritesh	Rafikbhai	Mahmand	Male	12-JAN-98	02-MAR-19	11000	pritesh@gmail.com	9612345678	Peon
8	Darshan	Maheshbhai	Patel	Male	01-DEC-98	02-JUL-14	10000	darshan@gmail.com	9634125678	Manager
9	Soham	Sahilbhai	Vyas	Male	26-FEB-01	22-SEP-19	38500	soham@gmail.com	8522345678	Peon
10	Pinank	Parimalbhai	Thummar	Male	01-JAN-00	02-AUG-18	10000	p123@gmail.com	9612632678	Security

10 rows returned in 0.00 seconds CSV Export

20. Create Simple Cursor and display all those Employee who are having Birthday today with birthday wish. (Hint: make use of FETCH INTO)

INPUT:

```
DECLARE
  CURSOR curs_bdy IS
    SELECT FName, MName, LName, DOB
    FROM EmployeeMst;

  vFName EmployeeMst.FName%TYPE;
  vMName EmployeeMst.MName%TYPE;
  vLName EmployeeMst.LName%TYPE;
```

```
  vDOB  EmployeeMst.DOB%TYPE;
BEGIN
  OPEN curs_bdy;
  IF curs_bdy%ISOPEN THEN
    LOOP
      FETCH curs_bdy INTO vFName, vMName, vLName, vDOB;
      EXIT WHEN curs_bdy%NOTFOUND;

      IF TO_CHAR(vDOB, 'DD-MON') = '26-FEB' THEN
        DBMS_OUTPUT.PUT_LINE('Happy birthday ' || vFName || '.');
      END IF;
    END LOOP;
  END IF;

  CLOSE curs_bdy; -- Close the cursor after usage
END;
```

OUTPUT:

```
Happy birthday Ritu.
Happy birthday Soham.

Statement processed.
```

21. Create Cursor for Loop and display all those Employee who have joined in month of July.

INPUT:

```
DECLARE
CURSOR curs_doj IS
SELECT *
FROM EmployeeMst;
i number(10);
BEGIN
FOR i IN curs_doj
LOOP
IF TO_CHAR(i.DOJ,'MON')='JUL' THEN
DBMS_OUTPUT.PUT_LINE('the employee :- '||' '||i.FName||' '||i.MName||' '||i.LName);
```

```
END IF;
END LOOP;
COMMIT;
END;
```

OUTPUT:

```
the employee :-  Shruti Vipulbhai Patel
the employee :-  Darshan Maheshbhai Patel

Statement processed.
```

22. Create Cursor for loop which Display all those employees who lives in Surat city.

INPUT:

```
DECLARE
  CURSOR curs_city IS
    SELECT EID, FName, MName, LName, Address
    FROM EmployeeMst;
  vCity VARCHAR(20);
BEGIN
  FOR i IN curs_city LOOP
    IF i.Address.City ='Surat' THEN
      DBMS_OUTPUT.PUT_LINE('Employee ' || i.FName || ' lives in ' || i.Address.City);
    END IF;
  END LOOP;
END;
```

OUTPUT:

```
Employee Pallavi lives in Surat
Employee Palak lives in Surat
Employee Arun lives in Surat
Employee Shruti lives in Surat

Statement processed.
```

23. Create PL/SQL Block to identify in-Built Exception and Give Proper Message for it.

INPUT:

```
DECLARE
  v_employee_name VARCHAR(50);
  v_employee_id NUMBER;
BEGIN
v_employee_id:=:v_employee_id;
  SELECT FName INTO v_employee_name
  FROM EmployeeMst
  WHERE EID = v_employee_id;
  DBMS_OUTPUT.PUT_LINE('Employee Name: ' || v_employee_name);
EXCEPTION
  WHEN NO_DATA_FOUND THEN
    DBMS_OUTPUT.PUT_LINE('Employee with ID ' || v_employee_id || ' not found.');
  WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('An error occurred.');
END;
```

OUTPUT:

```
Employee Name: Pinank

Statement processed.
```

24. Create PL/SQL Block to create user named Exception and Give Proper Message for it.

INPUT:

```
DECLARE
  user_defined_exception EXCEPTION;
  PRAGMA EXCEPTION_INIT(user_defined_exception, -20001);
  v_variable NUMBER := 10;
BEGIN
  IF v_variable > 5 THEN
    RAISE user_defined_exception;
  END IF;
EXCEPTION
  WHEN user_defined_exception THEN
    DBMS_OUTPUT.PUT_LINE('Custom Exception: Variable is greater than 5.');
END;
```

OUTPUT:

```
Custom Exception: Variable is greater than 5.

Statement processed.
```

25. Create PL/SQL Block to create user defined Exception and Give Proper Message for it.

INPUT:

```
DECLARE
  x INT;
  y INT;
  div_r FLOAT;
```

```
  exp1 EXCEPTION;

BEGIN
  x := :x;
  y := :y;

  IF (y = 0) THEN
    -- Raise the user-defined exception if y is zero
    RAISE exp1;
  ELSE
    div_r := x / y;
    DBMS_OUTPUT.PUT_LINE('The value of ' || x || ' divided by ' || y || ' is ' || div_r);
  END IF;

EXCEPTION
  WHEN exp1 THEN
    DBMS_OUTPUT.PUT_LINE('Error: Division by zero is not allowed');
END;
```

OUTPUT:

```
Error: Division by zero is not allowed

Statement processed.
```

26. Create user defined Procedure for summation and make use of it.

INPUT:

```
CREATE OR REPLACE PROCEDURE FUN_SUM( x IN NUMBER, y IN NUMBER, z OUT NUMBER)
IS
BEGIN
z:=x+y;
END;

DECLARE
a NUMBER;
b NUMBER;
c NUMBER;
BEGIN
a:=:a;
b:=:b;
FUN_SUM(a,b,c);
```

```
DBMS_OUTPUT.PUT_LINE('THE SUMMATION OF '||a||' AND '||b||' IS '||c);
END;
```

OUTPUT:

```
Procedure created.

THE SUMMATION OF 5 AND 2 IS 7

Statement processed.
```

27. Create user defined Function for summation and make use of it.

INPUT:

```
CREATE OR REPLACE FUNCTION SUMMATION(X IN NUMBER, Y IN NUMBER,Z OUT
NUMBER)
RETURN NUMBER IS
BEGIN
Z:=X+Y;
RETURN Z;
END;


DECLARE
A NUMBER:=10;
B NUMBER:=12;
C NUMBER;
BEGIN
C:=SUMMATION(A,B,C);
DBMS_OUTPUT.PUT_LINE('SUMMATION OF A AND B IS  '||C);
END;
```

OUTPUT:

```
Function created.
```

```
SUMMATION OF A AND B IS  22

Statement processed.
```

28. Create user defined Procedure to check whether the given number is even or odd.

INPUT:

```
CREATE OR REPLACE PROCEDURE FUN_EVENODD( x IN NUMBER) AS
R NUMBER;
BEGIN
R:=x MOD 2;
IF R=0 THEN
DBMS_OUTPUT.PUT_LINE(x ||' IS AN EVEN NUMBER');
ELSE
DBMS_OUTPUT.PUT_LINE(x||' IS A ODD NUMBER');
END IF;
END;

DECLARE
X NUMBER;
BEGIN
X:=:X;
FUN_EVENODD(X);
END;
```

OUTPUT:

```
Procedure created.

5 IS A ODD NUMBER

Statement processed.
```

29. Create user defined function to check whether the given year is leap year or not.

INPUT:

```
CREATE OR REPLACE FUNCTION Leap_yr(a IN NUMBER)
RETURN NUMBER IS
flag NUMBER;
BEGIN
IF a MOD 4=0 THEN
flag:=1;
ELSE
flag:=0;
END IF;
RETURN flag;
END;

DECLARE
X NUMBER;
flag NUMBER;
BEGIN
X:=:X;
flag:=Leap_yr(X);
IF flag=1 then
dbms_output.put_line(X ||' is a leap year');
ELSE
dbms_output.put_line(X ||' is not a leap year');
END IF;
END;
```

OUTPUT:

```
Function created.

2004 is a leap year

Statement processed.
```