DECLARE

   i number(1);

   j number(1);

BEGIN

   << outer\_loop >>

   FOR i IN 1..3 LOOP

      << inner\_loop >>

      FOR j IN 1..3 LOOP

         dbms\_output.put\_line('i is: '|| i || ' and j is: ' || j);

      END loop inner\_loop;

   END loop outer\_loop;

END;

/

DECLARE

   a integer :=4;

   b integer :=5;

   c integer;

BEGIN

   c:=a+b;

   dbms\_output.put\_line('sum of '||a||' and '||b||' is = '||c);

END;

/

 DECLARE

   a number;

PROCEDURE squareNum(x IN OUT number) IS

BEGIN

  x := x \* x;

END;

BEGIN

   a:= 23;

   squareNum(a);

   dbms\_output.put\_line(' Square of (23): ' || a);

END;

/

DECLARE

num number;

factorial number;

FUNCTION fact(x number)

RETURN number

IS

f number;

BEGIN

 IF x=0 THEN

 f:=1;

 ELSE

 f:=x\*fact(x-1);

 END IF;

 RETURN f;

 END;

 BEGIN

 num:=6;

 factorial:=fact(num);

 dbms\_output.put\_line('factorial of '||num||' is '||factorial);

 END;

 /

DECLARE

   a number;

   b number;

   c number;

PROCEDURE findMin(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:= 23;

   b:= 45;

   findMin(a, b, c);

   dbms\_output.put\_line(' Minimum of (23, 45) : ' || c);

END;

/

DECLARE

    var varchar2(40) := 'Hello World';

BEGIN

    dbms\_output.put\_line(var);

END;

/

DECLARE

   a integer :=4;

   b integer :=5;

   c integer;

BEGIN

   c:=a+b;

   dbms\_output.put\_line('sum of '||a||' and '||b||' is = '||c);

END;

/

DECLARE

   type namesarray IS VARRAY(5) OF VARCHAR2(10);

   type grades IS VARRAY(5) OF INTEGER;

   names namesarray;

   marks grades;

   total integer;

BEGIN

   names := namesarray('Kavita', 'Pritam', 'Ayan', 'Rishav', 'Aziz');

   marks:= grades(98, 97, 78, 87, 92);

   total := names.count;

   dbms\_output.put\_line('Total '|| total || ' Students');

   FOR i in 1 .. total LOOP

      dbms\_output.put\_line('Student: ' || names(i) || '

      Marks: ' || marks(i));

   END LOOP;

END;

/

CREATE TABLE CUSTOMERS(

   ID   INT NOT NULL,

   NAME VARCHAR (20) NOT NULL,

   AGE INT NOT NULL,

   ADDRESS CHAR (25),

   SALARY   DECIMAL (18, 2),

   PRIMARY KEY (ID)

);

INSERT INTO CUSTOMERS (ID,NAME,AGE,ADDRESS,SALARY)

VALUES (1, 'Ramesh', 32, 'Ahmedabad', 2000.00 );

INSERT INTO CUSTOMERS (ID,NAME,AGE,ADDRESS,SALARY)

VALUES (2, 'Khilan', 25, 'Delhi', 1500.00 );

INSERT INTO CUSTOMERS (ID,NAME,AGE,ADDRESS,SALARY)

VALUES (3, 'kaushik', 23, 'Kota', 2000.00 );

INSERT INTO CUSTOMERS (ID,NAME,AGE,ADDRESS,SALARY)

VALUES (4, 'Chaitali', 25, 'Mumbai', 6500.00 );

INSERT INTO CUSTOMERS (ID,NAME,AGE,ADDR);ESS,SALARY)

VALUES (5, 'Hardik', 27, 'Bhopal', 8500.0);

 DECLARE

   total\_rows number(2);

BEGIN

   UPDATE customers

   SET salary = salary + 500;

   IF sql%notfound THEN

      dbms\_output.put\_line('no customers selected');

   ELSIF sql%found THEN

      total\_rows := sql%rowcount;

      dbms\_output.put\_line( total\_rows || ' customers selected ');

   END IF;

END;

/

INSERT INTO CUSTOMERS (ID,NAME,AGE,ADDRESS,SALARY)

VALUES (6, 'Komal', 22, 'MP', 4500.00 );

DECLARE

   c\_id customers.id%type;

   c\_name customers.name%type;

   c\_addr customers.address%type;

   CURSOR c\_customers is

      SELECT id, name, address FROM customers;

BEGIN

   OPEN c\_customers;

   LOOP

   FETCH c\_customers into c\_id, c\_name, c\_addr;

      EXIT WHEN c\_customers%notfound;

      dbms\_output.put\_line(c\_id || ' ' || c\_name || ' ' || c\_addr);

   END LOOP;

   CLOSE c\_customers;

END;

/

CREATE OR REPLACE TRIGGER display\_salary\_changes

BEFORE DELETE OR INSERT OR UPDATE ON customers

FOR EACH ROW

WHEN (NEW.ID > 0)

DECLARE

   sal\_diff number;

BEGIN

   sal\_diff := :NEW.salary  - :OLD.salary;

   dbms\_output.put\_line('Old salary: ' || :OLD.salary);

   dbms\_output.put\_line('New salary: ' || :NEW.salary);

   dbms\_output.put\_line('Salary difference: ' || sal\_diff);

END;

/