DECLARE

x integer := 10;

y integer := 20;

r integer;

BEGIN

r := x+y;

dbms\_output.put\_line('Sum of x , y is :' || r);

END;

DECLARE

x integer := 10;

y integer := 20;

r integer;

BEGIN

r := x+y;

IF r = 30 THEN

RAISE INVALID\_NUMBER;

ELSE

dbms\_output.put\_line('Sum of x , y is :' || r);

END IF;

EXCEPTION

WHEN INVALID\_NUMBER THEN

dbms\_output.put\_line('Caught an exception thwon by self');

WHEN others THEN

dbms\_output.put\_line('Genral Exception Received');

END;

DECLARE

x integer := 10;

y integer := 20;

r integer;

BEGIN

r := x+y;

IF r = 30 THEN

RAISE INVALID\_NUMBER;

ELSE

dbms\_output.put\_line('Sum of x , y is :' || r);

END IF;

EXCEPTION

WHEN INVALID\_NUMBER THEN

dbms\_output.put\_line('Caught an exception thwon by self');

WHEN others THEN

dbms\_output.put\_line('Genral Exception Received');

END;

DECLARE

x integer := 10;

y integer := 20;

r integer;

BEGIN

r := x+y;

IF r = 0 THEN

RAISE INVALID\_NUMBER;

ELSE

dbms\_output.put\_line('Sum of x , y is :' || r);

END IF;

EXCEPTION

WHEN INVALID\_NUMBER THEN

dbms\_output.put\_line('Caught an exception thwon by self');

WHEN others THEN

dbms\_output.put\_line('Genral Exception Received'|| r);

END;

DECLARE

msg varchar2(100) := 'Welcome to India';

BEGIN

dbms\_output.put\_line('msg is :' ||msg);

END;

DECLARE

msg varchar2(100) := 'Welcome to India';

BEGIN

dbms\_output.put\_line('msg is :' ||msg);

dbms\_output.put\_line('msg is :' ||UPPER(msg));

END;

DECLARE

msg varchar2(100) := 'Welcome to India';

BEGIN

dbms\_output.put\_line('msg is :' ||msg);

dbms\_output.put\_line('msg is :' ||UPPER(msg));

dbms\_output.put\_line('msg is :' ||LOWER(msg));

END;

DECLARE

msg varchar2(100) := 'Welcome to India';

BEGIN

dbms\_output.put\_line('msg is :' ||msg);

dbms\_output.put\_line('msg is :' ||UPPER(msg));

dbms\_output.put\_line('msg is :' ||LOWER(msg));

dbms\_output.put\_line('msg is :' ||INITCAP(msg));

END;

DECLARE

msg varchar2(100) := 'Welcome to India';

BEGIN

dbms\_output.put\_line('msg is :' ||msg);

dbms\_output.put\_line('msg is :' ||UPPER(msg));

dbms\_output.put\_line('msg is :' ||LOWER(msg));

dbms\_output.put\_line('msg is :' ||INITCAP(msg));

dbms\_output.put\_line('Lenghth of the msg is :' ||LENGTH(msg));

END;

DECLARE

msg varchar2(100) := 'Welcome to India';

BEGIN

dbms\_output.put\_line('msg is :' ||msg);

dbms\_output.put\_line('msg is :' ||UPPER(msg));

dbms\_output.put\_line('msg is :' ||LOWER(msg));

dbms\_output.put\_line('msg is :' ||INITCAP(msg));

dbms\_output.put\_line('Lenghth of the msg is :' ||LENGTH(msg));

dbms\_output.put\_line('Replace ex is :' ||UPPER(msg,India,USA));

END;

DECLARE

msg varchar2(100) := 'Welcome to India';

BEGIN

dbms\_output.put\_line('msg is :' ||msg);

dbms\_output.put\_line('msg is :' ||UPPER(msg));

dbms\_output.put\_line('msg is :' ||LOWER(msg));

dbms\_output.put\_line('msg is :' ||INITCAP(msg));

dbms\_output.put\_line('Lenghth of the msg is :' ||LENGTH(msg));

dbms\_output.put\_line('Replace ex is :' ||UPPER(msg,"India","USA"));

END;

DECLARE

msg varchar2(100) := 'Welcome to India';

BEGIN

dbms\_output.put\_line('msg is :' ||msg);

dbms\_output.put\_line('msg is :' ||UPPER(msg));

dbms\_output.put\_line('msg is :' ||LOWER(msg));

dbms\_output.put\_line('msg is :' ||INITCAP(msg));

dbms\_output.put\_line('Lenghth of the msg is :' ||LENGTH(msg));

dbms\_output.put\_line('Replace ex is :' ||REPLACE(msg,"India","USA"));

END;

DECLARE

msg varchar2(100) := 'Welcome to India';

BEGIN

dbms\_output.put\_line('msg is :' ||msg);

dbms\_output.put\_line('msg is :' ||UPPER(msg));

dbms\_output.put\_line('msg is :' ||LOWER(msg));

dbms\_output.put\_line('msg is :' ||INITCAP(msg));

dbms\_output.put\_line('Lenghth of the msg is :' ||LENGTH(msg));

dbms\_output.put\_line('Replace ex is :' ||REPLACE(msg,'India','USA'));

END;

DECLARE

TYPE namearray IS VARRAY(5) OF VARCHAR(100);

TYPE agearray IS VARRAY(5) OF INTEGER;

name namearray;

age agearray;

BEGIN

name := namearray('Rohit','Rajesh','Vijay','Bala','Chadra');

age := agearray(37,30,20,21,22);

END;

DECLARE

TYPE namearray IS VARRAY(5) OF VARCHAR(100);

TYPE agearray IS VARRAY(5) OF INTEGER;

name namearray;

age agearray;

BEGIN

name := namearray('Rohit','Rajesh','Vijay','Bala','Chadra');

age := agearray(37,30,20,21,22);

dbms\_output.put\_line('name : '|| name(1));

END;

DECLARE

TYPE namearray IS VARRAY(5) OF VARCHAR(100);

TYPE agearray IS VARRAY(5) OF INTEGER;

name namearray;

age agearray;

BEGIN

name := namearray('Rohit','Rajesh','Vijay','Bala','Chadra');

age := agearray(37,30,20,21,22);

dbms\_output.put\_line('name : '|| name(1) || 'age:' || age(1));

END;

DECLARE

TYPE namearray IS VARRAY(5) OF VARCHAR(100);

TYPE agearray IS VARRAY(5) OF INTEGER;

name namearray;

age agearray;

BEGIN

name := namearray('Rohit','Rajesh','Vijay','Bala','Chadra');

age := agearray(37,30,20,21,22);

dbms\_output.put\_line('name : '|| name(1) || ' age:' || age(1));

END;

CREATE OR REPLACE PROCEDURE sampleprocedue

AS

BEGIN

dbms\_output.put\_line('Hello Everyone!');

END;

CREATE OR REPLACE PROCEDURE sampleprocedue

AS

BEGIN

dbms\_output.put\_line('Hello Everyone!');

END;

BEGIN

sampleprocedue

END;

DROP PROCEDURE sampleprocedue

PROCEDUE findmax(x IN number, y IN number, z OUT number)

IS

BEGIN

IF x > y THEN

z := x;

ELSE

Z := y;

END IF;

END;

BEGIN

findmax(10,20,r);

dbms\_output.put\_line('Max value is '|| r);

END;

DECLARE

r number;

PROCEDUE findmax(x IN number, y IN number, z OUT number)

IS

BEIN

IF x > y THEN

z := x;

ELSE

z := y;

END IF;

END;

BEGIN

findmax(10,20,r);

dbms\_output.put\_line('Max value is '|| r);

END;

DECLARE

r number;

PROCEDUE findmax(x IN number, y IN number, z OUT number)

IS

BEGIN

IF x > y THEN

z := x;

ELSE

z := y;

END IF;

END;

BEGIN

findmax(10,20,r);

dbms\_output.put\_line('Max value is '|| r);

END;

DECLARE

r number;

PROCEDUE findmax(x IN number, y IN number, z OUT number)

IS

BEGIN

IF x > y THEN

z := x;

ELSE

z := y;

END IF;

END;

DECLARE

r number;

PROCEDUE findmax(x IN number, y IN number, z OUT number)

IS

BEGIN

IF x > y THEN

z := x;

ELSE

z := y;

END IF;

END;

BEGIN

findmax(10,20,r);

dbms\_output.put\_line('Max value is '|| r);

END;

DECLARE

r number;

PROCEDURE findmax(x IN number, y IN number, z OUT number)

IS

BEGIN

IF x > y THEN

z := x;

ELSE

z := y;

END IF;

END;

BEGIN

findmax(10,20,r);

dbms\_output.put\_line('Max value is '|| r);

END;

DECLARE

r number;

PROCEDURE findmax(x IN number, y IN number, z OUT number)

IS

BEGIN

IF x > y THEN

z := x;

ELSE

z := y;

END IF;

END;

BEGIN

findmax(10,20,r);

dbms\_output.put\_line('Max value is '|| r);

END;

DECLARE

a number;

PROCEDURE squareit(x IN OUT number) IS

BEGIN

x := x\*x;

END;

BEGIN

a:=10;

squareit(a)

END;

DECLARE

a number;

PROCEDURE squareit(x IN OUT number) IS

BEGIN

x := x\*x;

END;

BEGIN

a := 10;

squareit(a);

dbms\_output.put\_line('SQUARE OF 10 IS : ' || a);

END;

INSERT INTO customer(id,name) VALUES(101,'Rohit')

CREATE TABLE customer(

id INT,

name VARCHAR(100)

)

INSERT INTO customer(id,name) VALUES(100,'Rohit')

INSERT INTO customer(id,name) VALUES(101,'Durga')

CREATE OR REPLACE FUNCTION sumnums

RETURN number IS

res number(2);

BEGIN

SELECT sum(id) into res

FROM customer;

RETURN res;

END;

DECLARE

r number(3);

BEGIN

r := sumnums();

dbms\_output.put\_line(' Sum of customer ID is : ' || r);

END;

DECLARE

a number;

b number;

c number;

FUNCTION findmax(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;

BEGIN

a := 10;

b := 20;

c := findmax(a,b);

dbms\_output.put\_line(' Max number is : ' || c);

END;

CREATE OR REPLACE TRIGEGR print\_custid

BEFORE UPDATE ON customer

FOR EACH ROW

WHEN (OLD.id >0)

DECLARE

x number

BEGIN

x := OLD.id;

dbms\_output.put\_line(x);

END;

CREATE OR REPLACE TRIGGER print\_custid

BEFORE UPDATE ON customer

FOR EACH ROW

WHEN (NEW.id >0)

DECLARE

x number;

BEGIN

x := :NEW.id - :OLD.id;

dbms\_output.put\_line('OLD ID :|| :OLD.id);

dbms\_output.put\_line('NEW ID :|| :NEW.id);

END;

UPDATE customer SET id=500 WHERE id=100

Create TABLE city\_audit 1)

empid NUMBER,

change\_date DATE,

old\_city VARCHAR2(100),

new\_city VARCHAR2(100)

)

CREATE OR REPLACE TRIGGER emp\_city\_audit

AFTER UPDATE OF CITY ON Employee

FOR EACH ROW

BEGIN

INSERT INTO city\_audit (empid, change\_date, old\_city, new\_City)

VALUES(:NEW.empid,SYSDATE,:OLD.city, :NEW.city);

END;

UPDATE Employee SET city='DELHI' WHERE empid=1002

SELECT \* FROM city\_audit