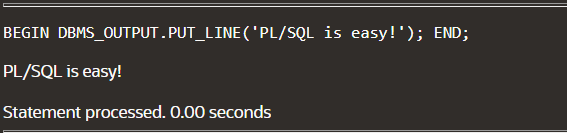
BEGIN

DBMS\_OUTPUT.PUT\_LINE('PL/SQL is easy!');

END;



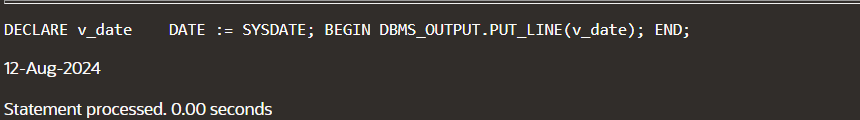
**DECLARE**

**v\_date DATE := SYSDATE;**

**BEGIN**

**DBMS\_OUTPUT.PUT\_LINE(v\_date);**

**END;**



DECLARE

v\_FIRST\_NAME VARCHAR2(35);

v\_LAST\_NAME VARCHAR2(20);

BEGIN

SELECT FIRST\_NAME,LAST\_NAME

INTO v\_FIRST\_NAME,v\_LAST\_NAME

FROM EMPLOYEES

WHERE LAST\_NAME ='priyadarshini';

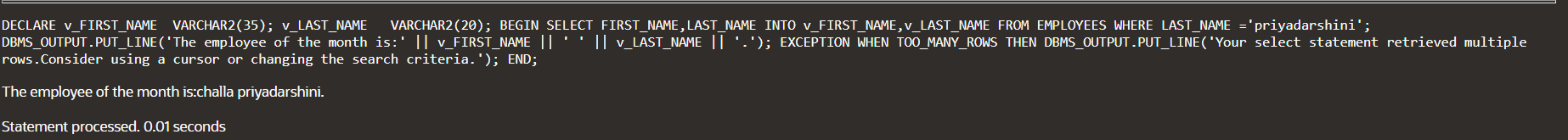
DBMS\_OUTPUT.PUT\_LINE('The employee of the month is:'

|| v\_FIRST\_NAME || ' ' || v\_LAST\_NAME || '.');

EXCEPTION

WHEN TOO\_MANY\_ROWS THEN

DBMS\_OUTPUT.PUT\_LINE('Your select statement retrieved multiple rows.Consider using a cursor or changing the search criteria.');

END;

DECLARE

a integer := 10;

b integer := 20;

c integer;

f real;

BEGIN

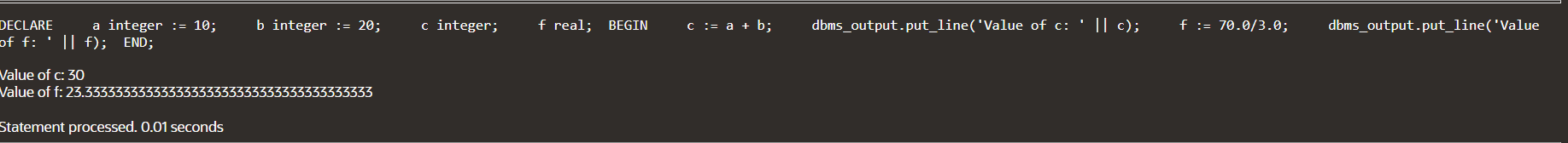
c := a + b;

dbms\_output.put\_line('Value of c: ' || c);

f := 70.0/3.0;

dbms\_output.put\_line('Value of f: ' || f);

END;



DECLARE

-- constant declaration

pi constant number := 3.141592654;

-- other declarations

radius number(5,2);

dia number(5,2);

circumference number(7, 2);

area number (10, 2);

BEGIN

-- processing

radius := 9.5;

dia := radius \* 2;

circumference := 2.0 \* pi \* radius;

area := pi \* radius \* radius;

-- output

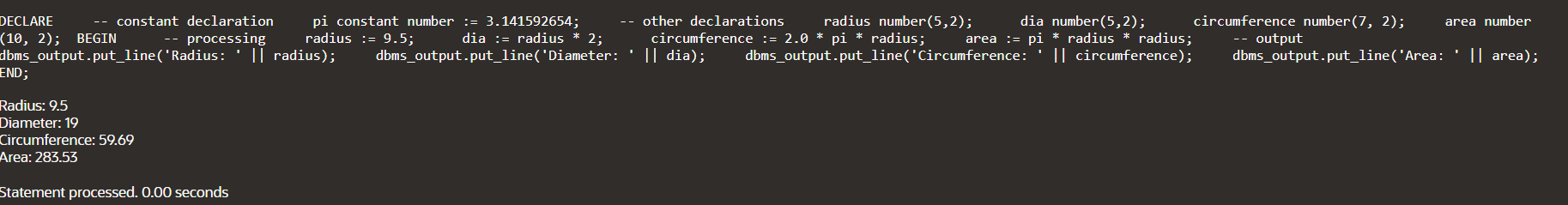
dbms\_output.put\_line('Radius: ' || radius);

dbms\_output.put\_line('Diameter: ' || dia);

dbms\_output.put\_line('Circumference: ' || circumference);

dbms\_output.put\_line('Area: ' || area);

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');

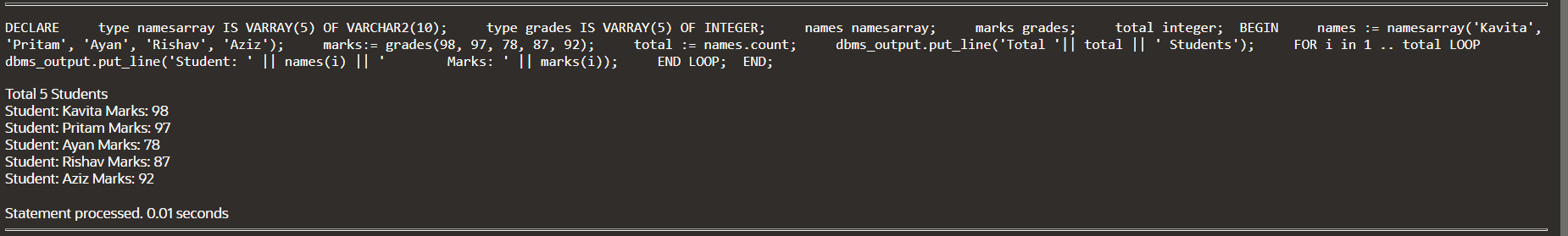
FOR i in 1 .. total LOOP

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

Marks: ' || marks(i));

END LOOP;

END;



DECLARE

v\_num NUMBER;

v\_even\_count NUMBER := 0;

v\_odd\_count NUMBER := 0;

BEGIN

FOR v\_num IN 1..10 LOOP

IF MOD(v\_num, 2) = 0 THEN

v\_even\_count := v\_even\_count + 1;

ELSE

v\_odd\_count := v\_odd\_count + 1;

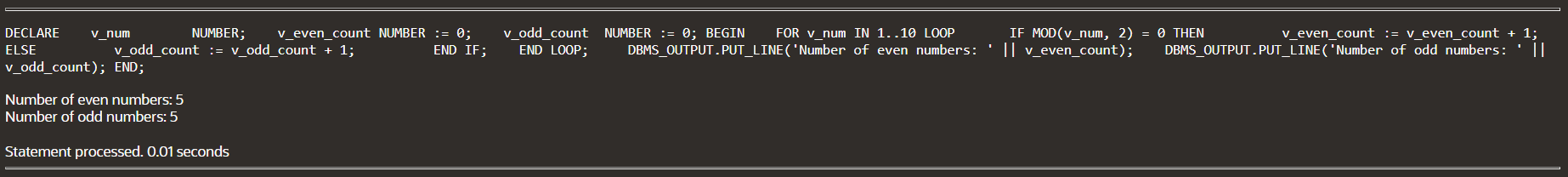
END IF;

END LOOP;

DBMS\_OUTPUT.PUT\_LINE('Number of even numbers: ' || v\_even\_count);

DBMS\_OUTPUT.PUT\_LINE('Number of odd numbers: ' || v\_odd\_count);

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

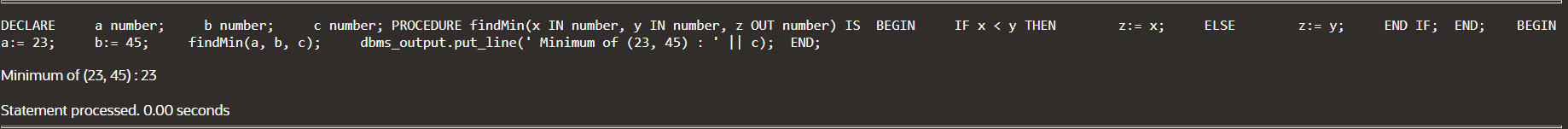
a:= 23;

b:= 45;

findMin(a, b, c);

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

END;



DECLARE

num1 NUMBER;

num2 NUMBER;

operation CHAR(1);

result NUMBER;

BEGIN

num1 := 10;

num2 := 5;

operation := '\*';

CASE operation

WHEN '+' THEN

result := num1 + num2;

WHEN '-' THEN

result := num1 - num2;

WHEN '\*' THEN

result := num1 \* num2;

WHEN '/' THEN

IF num2 = 0 THEN

DBMS\_OUTPUT.PUT\_LINE('Error: Division by zero');

RETURN;

ELSE

result := num1 / num2;

END IF;

ELSE

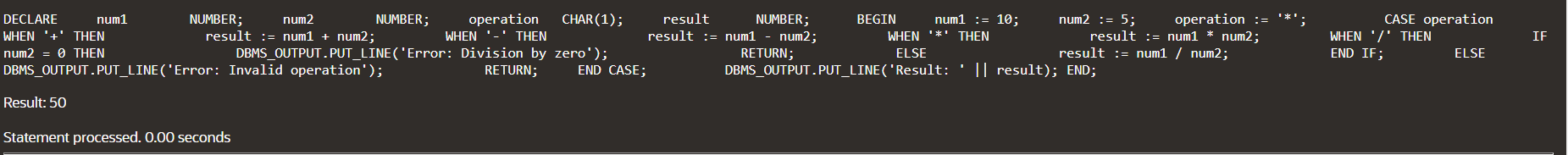
DBMS\_OUTPUT.PUT\_LINE('Error: Invalid operation');

RETURN;

END CASE;

DBMS\_OUTPUT.PUT\_LINE('Result: ' || result);

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;

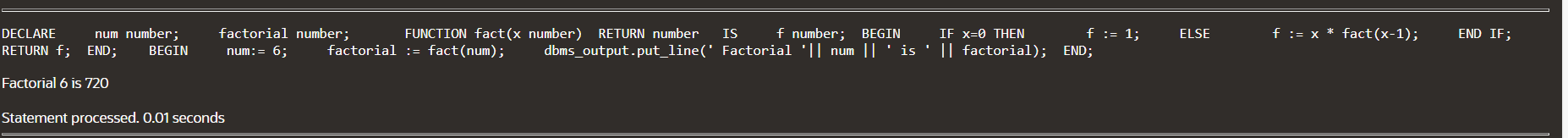
END;

BEGIN

num:= 6;

factorial := fact(num);

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

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

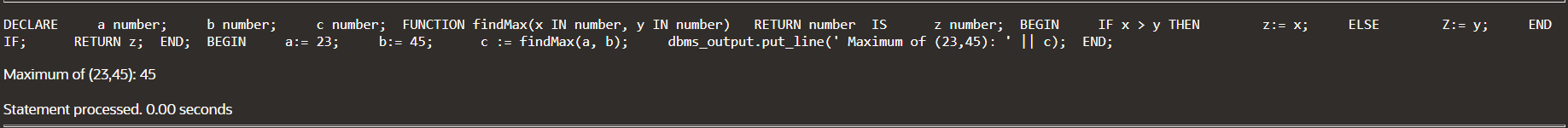
a:= 23;

b:= 45;

c := findMax(a, b);

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

END;



DECLARE

i NUMBER;

FUNCTION fibonacci(n NUMBER) RETURN NUMBER

IS

BEGIN

IF n <= 0 THEN

RETURN 0;

ELSIF n = 1 THEN

RETURN 1;

ELSE

RETURN fibonacci(n-1) + fibonacci(n-2);

END IF;

END fibonacci; -- add this to end the function declaration

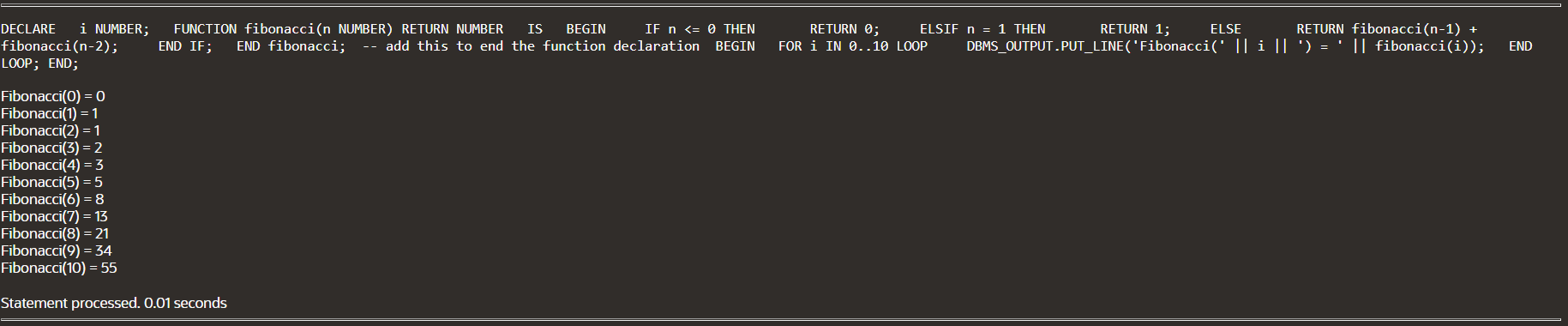
BEGIN

FOR i IN 0..10 LOOP

DBMS\_OUTPUT.PUT\_LINE('Fibonacci(' || i || ') = ' || fibonacci(i));

END LOOP;

END;



**IMPLICITY:**

DECLARE

total\_rows number(2);

BEGIN

UPDATE PERSON1

SET salary = salary + 500;

IF sql%notfound THEN

dbms\_output.put\_line('no PERSON1 selected');

ELSIF sql%found THEN

total\_rows := sql%rowcount;

dbms\_output.put\_line( total\_rows || ' PERSON1 selected ');

END IF;

END;

