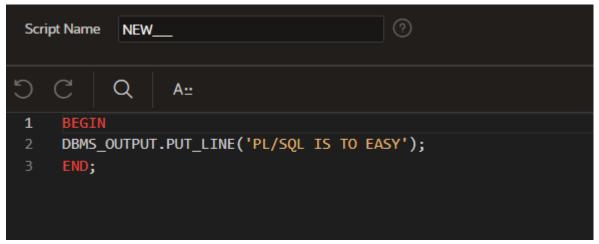
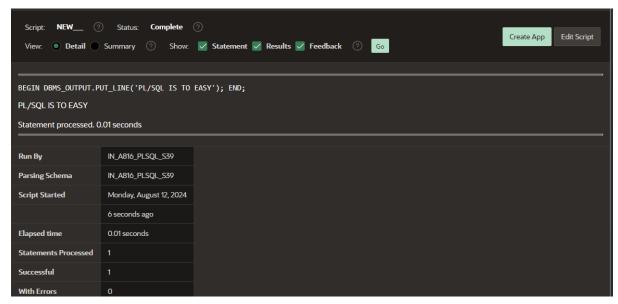
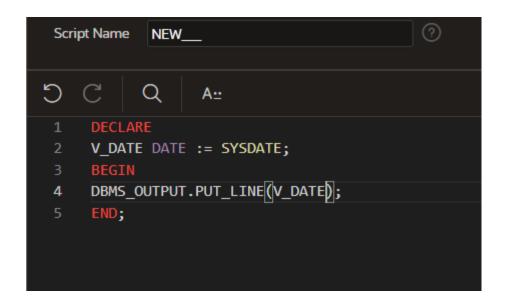
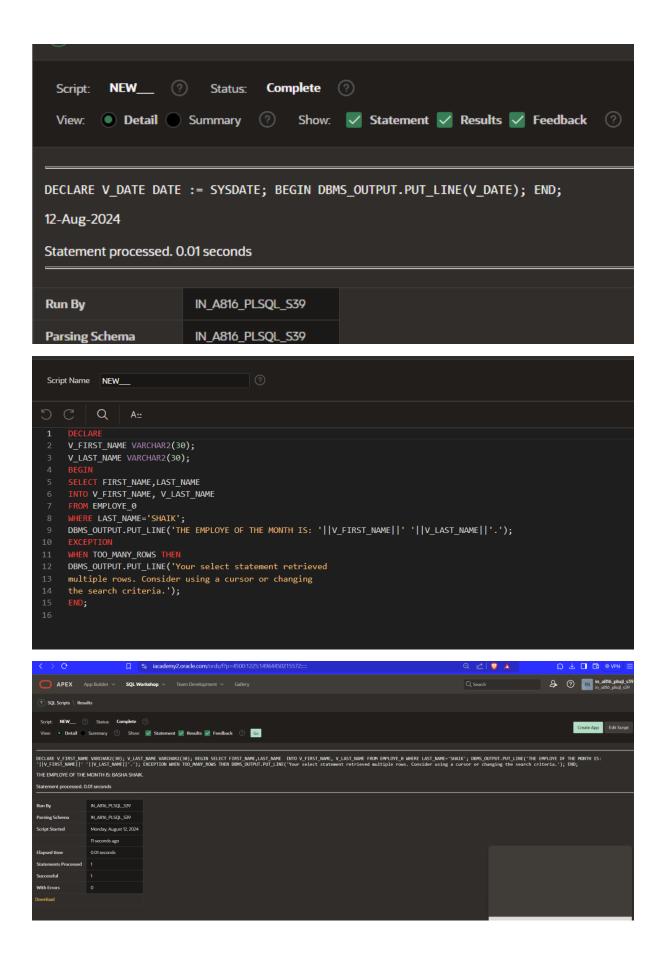
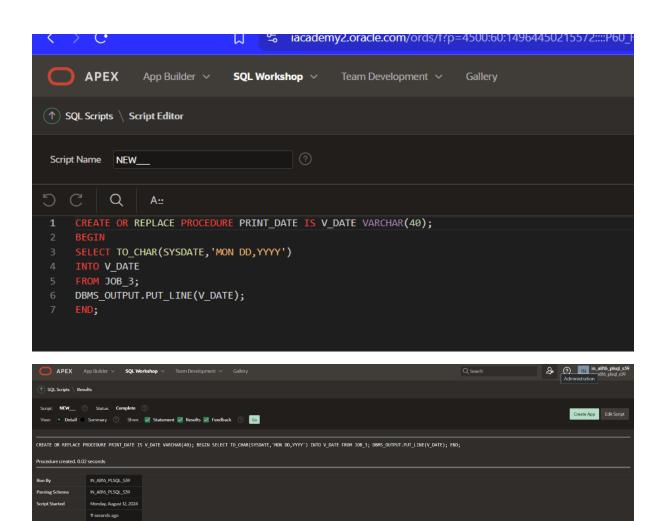
PL/SQL

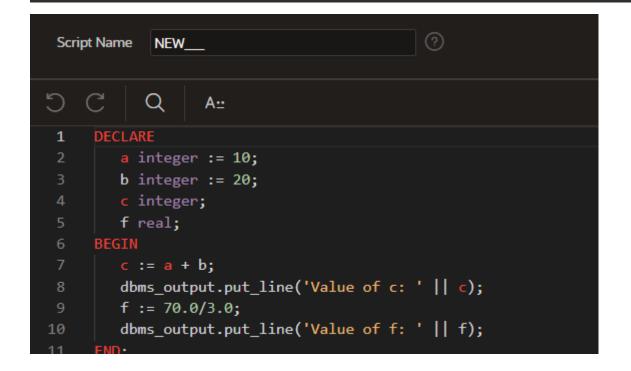




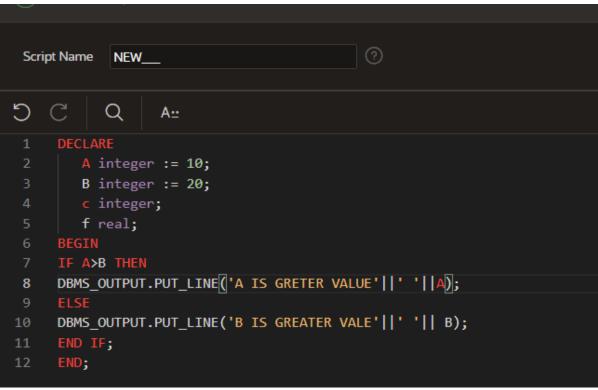






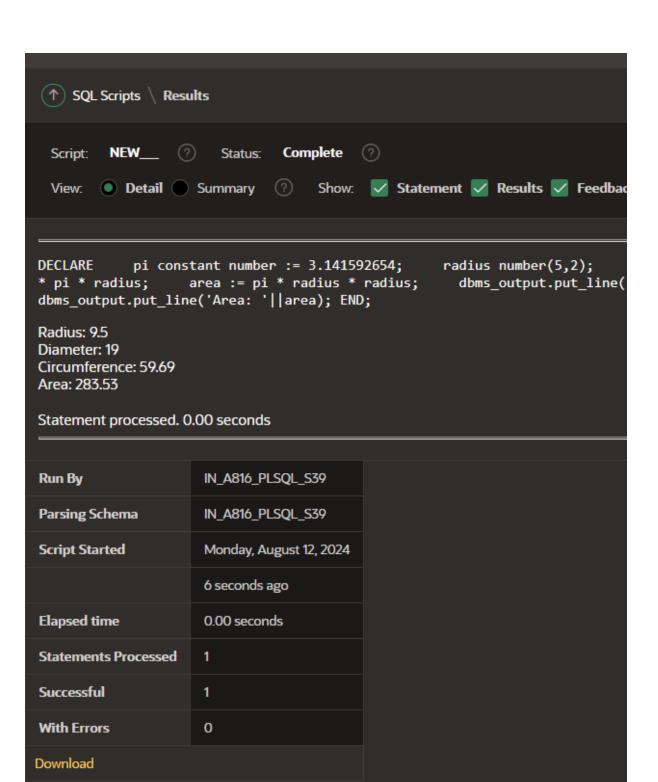


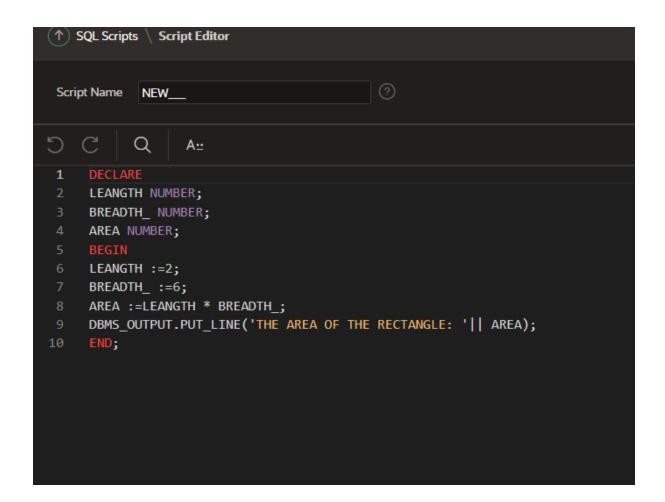


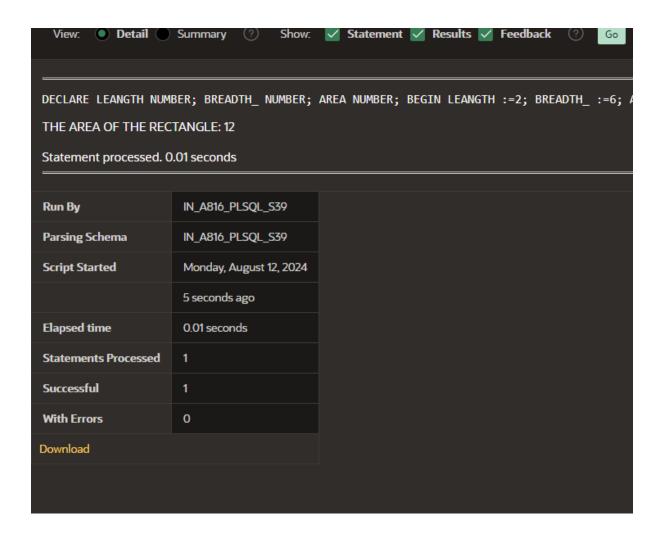


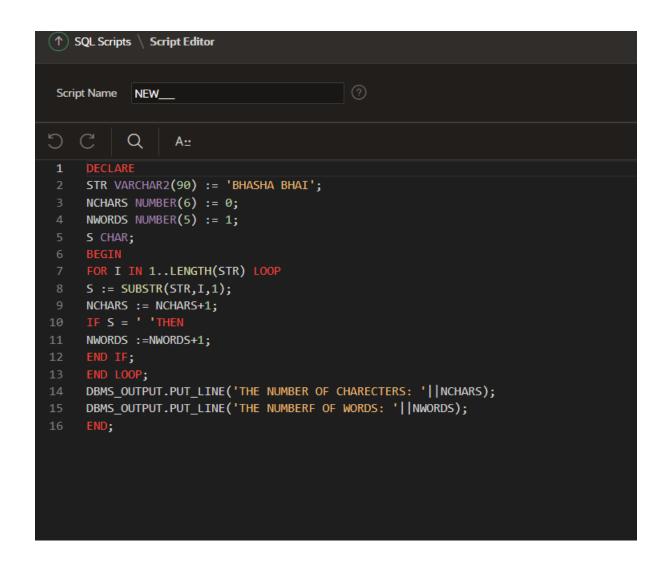


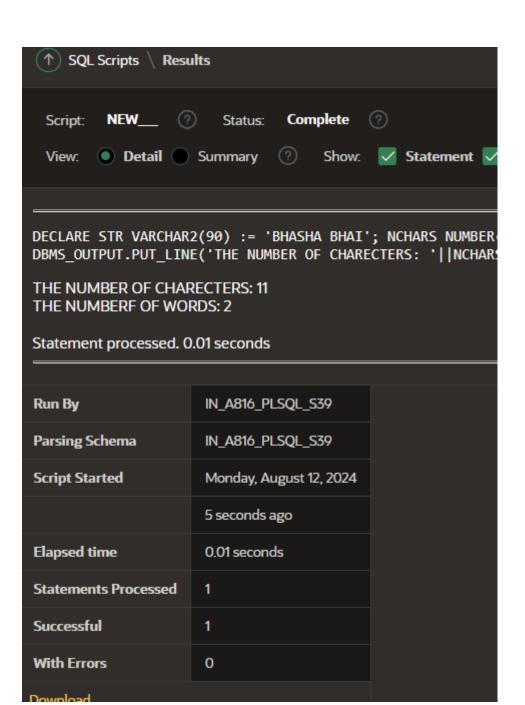
```
Script Name NEW___
           Q
                  A::
        pi constant number := 3.141592654;
        radius number(5,2);
        dia number(5,2);
        circumference number(7, 2);
        area number (10, 2);
8
        radius := 9.5;
        dia := radius * 2;
        circumference := 2.0 * pi * radius;
11
12
        area := pi * radius * radius;
        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;
```

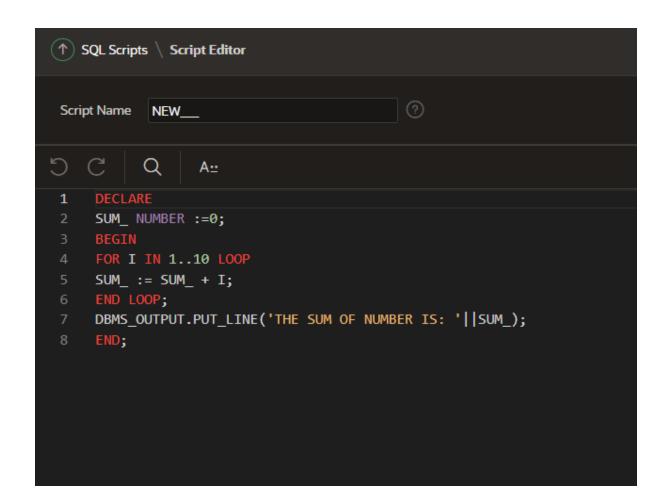


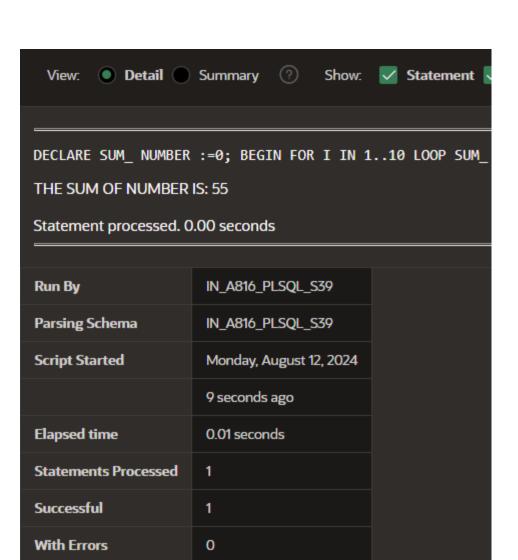






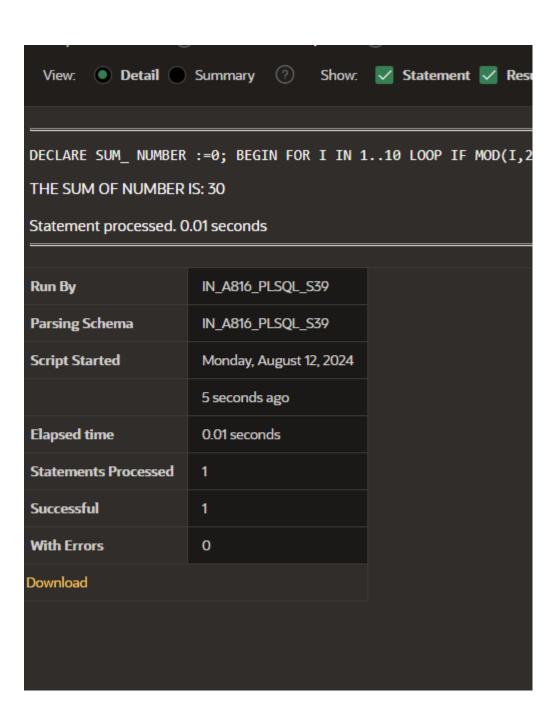






Download

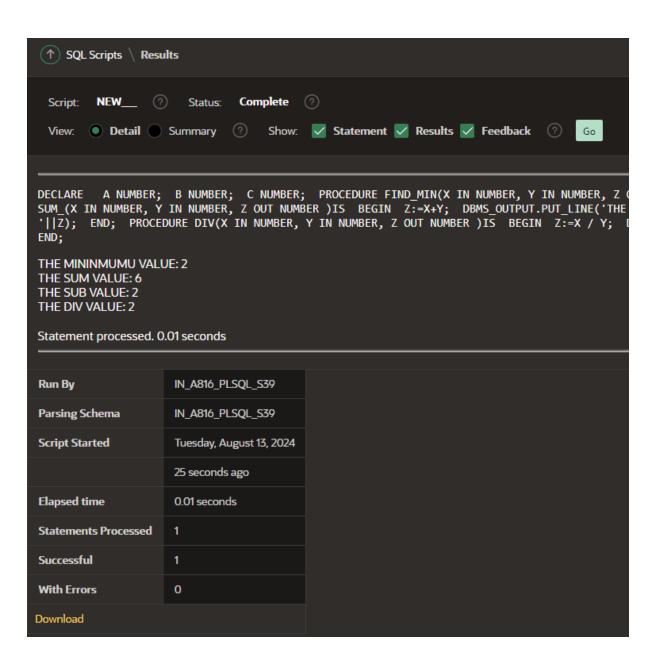
```
DECLARE
SUM_ NUMBER := 
Auto Complete - Ctrl+Space
BEGIN
FOR I IN 1..10 LOOP
IF MOD(I,2)=0 THEN
SUM_ := SUM_ + I;
END IF;
END LOOP;
DBMS_OUTPUT.PUT_LINE('THE SUM OF NUMBER IS: '||SUM_);
END;
```

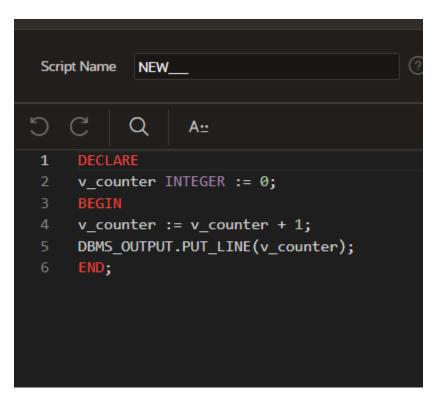


```
TYPE namesarray IS VARRAY(5) OF VARCHAR2(10);
        TYPE grades IS VARRAY(5) OF INTEGER;
        names namesarray;
        marks grades;
        total INTEGER;
        names := namesarray('Kavita', 'Pritam', 'Ayan', 'Rishav', 'Aziz');
        marks := grades(98, 97, 78, 87, 92);
        total := names.COUNT;
        DBMS_OUTPUT.PUT_LINE('Total ' || total || ' Students');
13
        FOR i IN 1 .. total LOOP
14
           IF marks(i) > 95 THEN
             DBMS_OUTPUT.PUT_LINE('A');
           ELSIF marks(i) <= 90 THEM
             DBMS_OUTPUT.PUT_LINE('B');
18
19
           DBMS_OUTPUT.PUT_LINE('Student: ' || names(i) || ' Marks: ' || marks(i));
20
21
        END LOOP;
```

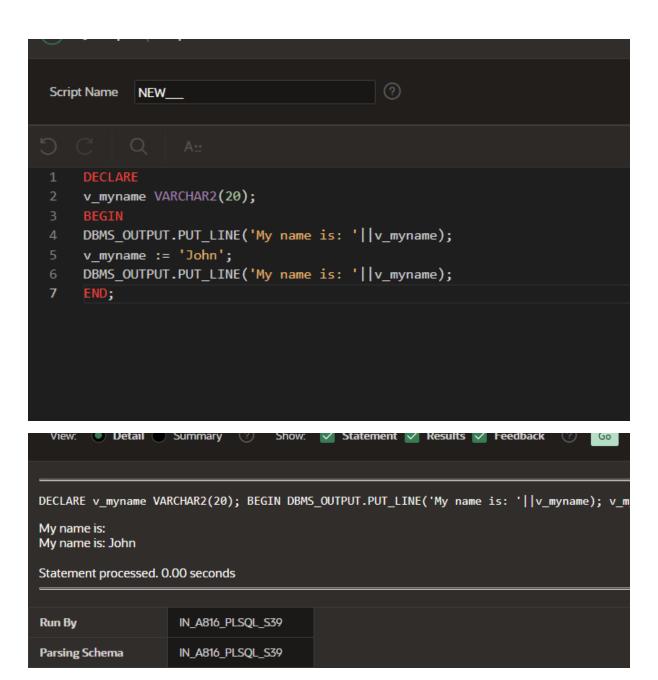


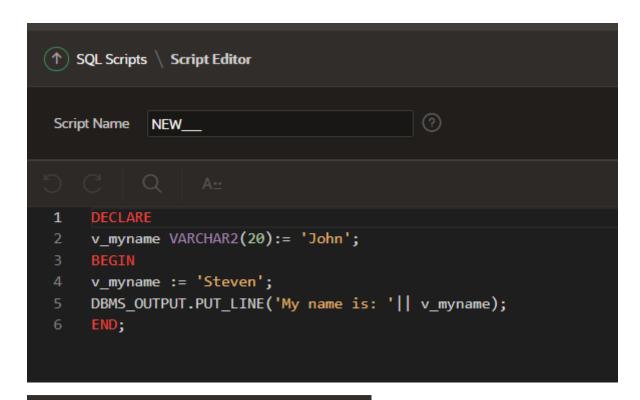
Script Name NEW_ Q A:: A NUMBER; B NUMBER; C NUMBER; PROCEDURE FIND_MIN(X IN NUMBER, Y IN NUMBER, Z OUT NUMBER) IS IF X<Y THEN Z:=X; Z:=Y; DBMS_OUTPUT.PUT_LINE('THE MININMUMU VALUE: '||Z); PROCEDURE SUM_(X IN NUMBER, Y IN NUMBER, Z OUT NUMBER)IS BEGIN Z:=X+Y;DBMS_OUTPUT.PUT_LINE('THE SUM VALUE: '||Z); PROCEDURE SUB(X IN NUMBER, Y IN NUMBER, Z OUT NUMBER)IS BEGIN Z:=X-Y; DBMS_OUTPUT.PUT_LINE('THE SUB VALUE: '||Z); END; PROCEDURE DIV(X IN NUMBER, Y IN NUMBER, Z OUT NUMBER)IS Z:=X / Y;DBMS_OUTPUT.PUT_LINE('THE DIV VALUE: '||Z); A:=4; B:=2; FIND_MIN(A,B,C); SUM_(A,B,C); SUB(A,B,C); DIV(A,B,C);







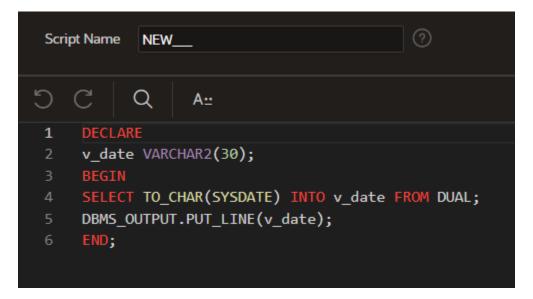


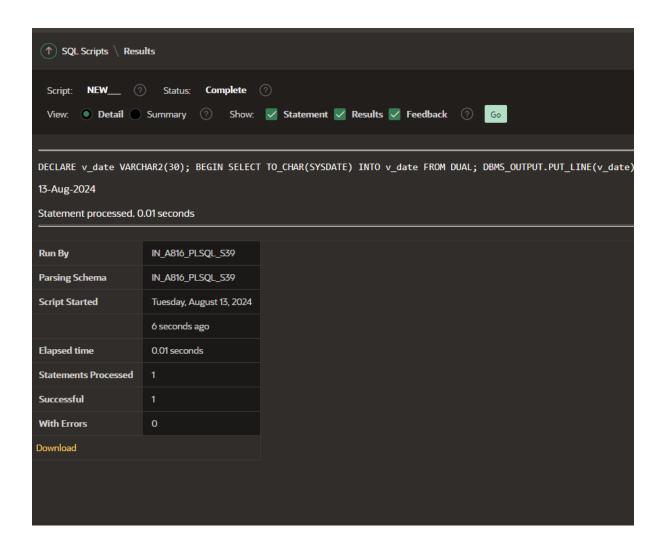


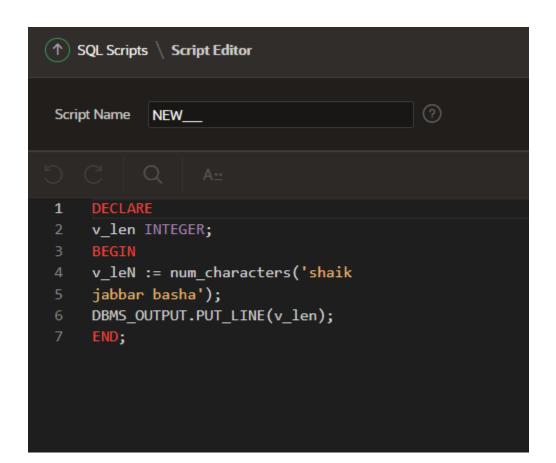
DECLARE v_myname VARCHAR2(20):= 'John';

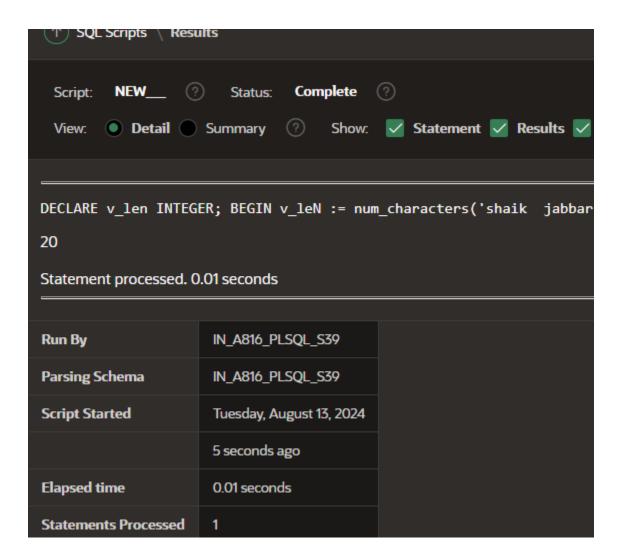
My name is: Steven

Statement processed. 0.01 seconds

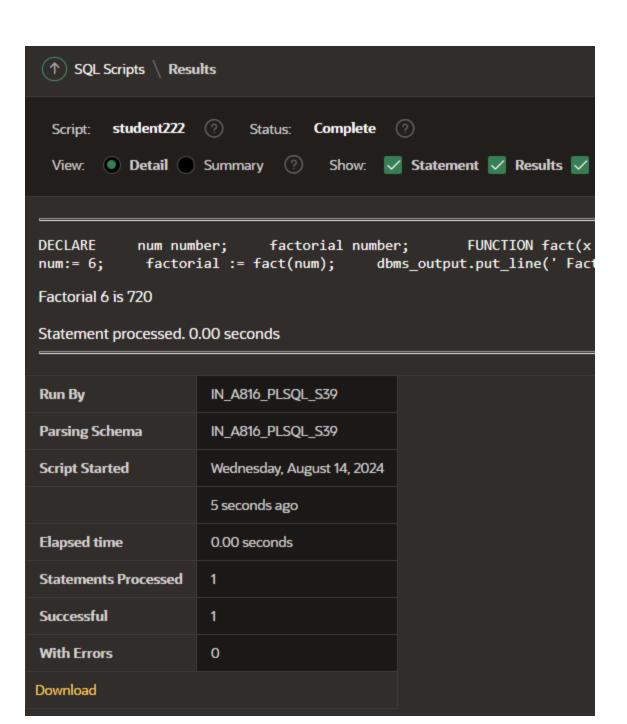




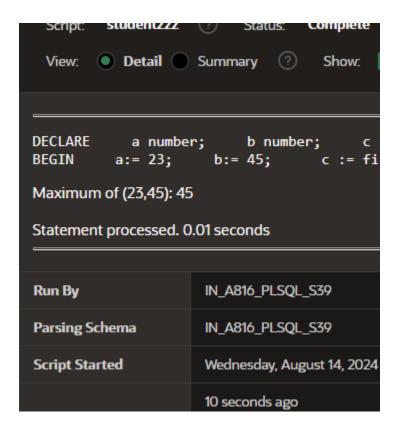




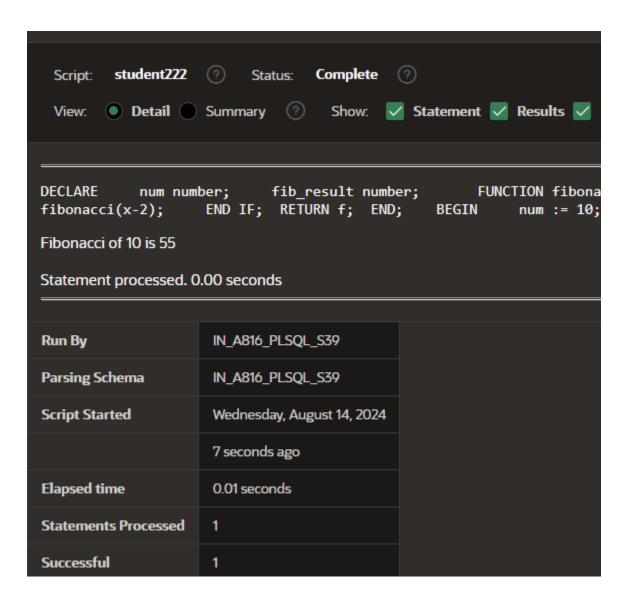
```
Script Name student222
         Q
                A::
       num number;
       factorial number;
    FUNCTION fact(x number)
    RETURN number
    f number;
      IF x=0 THEN
       f := 1;
       f := x * fact(x-1);
    RETURN f;
    END;
     num:= 6;
      factorial := fact(num);
      dbms_output.put_line(' Factorial '|| num || ' is ' || factorial);
    END;
```

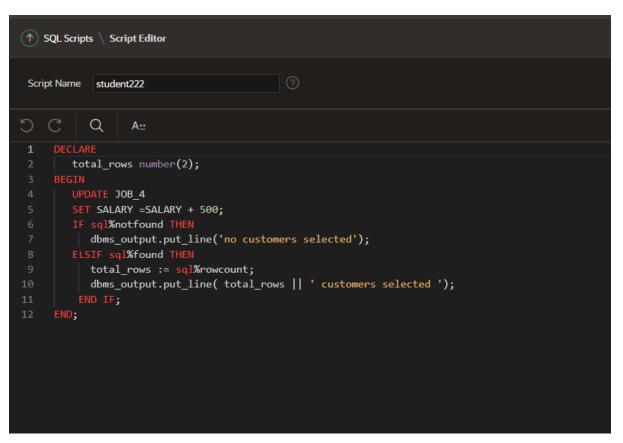


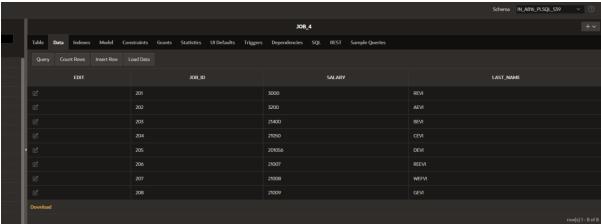
```
↑ SQL Scripts \ Script Editor
 Script Name
            student222
   C
           Q
                  A::
     DECLARE
 1
        a number;
        b number;
       c number;
     FUNCTION findMax(x IN number, y IN number)
     RETURN number
     IS
          z number;
        IF x > y THEN
11
         z:= x;
12
13
           Z:= y;
        END IF;
15
        RETURN z;
     END;
17
18
        a:= 23;
19
        b := 45;
        c := findMax(a, b);
        dbms_output.put_line(' Maximum of (23,45): ' || c);
21
22
     END;
```

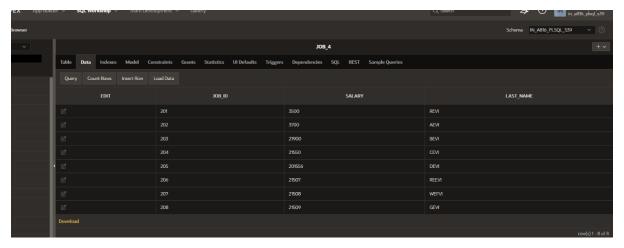


```
num number;
        fib_result number;
     FUNCTION fibonacci(x number)
     RETURN number
     f number;
        f := 0;
12
13
        f := 1;
14
15
        f := fibonacci(x-1) + fibonacci(x-2);
16
17
     RETURN f;
18
     END;
19
20
21
       num := 10;
22
       fib_result := fibonacci(num);
       dbms_output.put_line(' Fibonacci of ' || num || ' is ' || fib_result);
23
24
```









```
Script Name student222

Declare

EMP_ID EMP_LEMP_ID%type;

LAST_NAME EMP_LAST_NAME%type;

EMP_AD EMP_LEMP_AD%type;

CURSOR c_EMP_ is

SELECT EMP_ID, LAST_NAME, EMP_AD FROM EMP_;

BEGIN

OPEN c_EMP_;

LOOP

FETCH c_EMP_ into EMP_ID, LAST_NAME, EMP_AD;

EXIT WHEN c_EMP_%notfound;

dbms_output.put_line(EMP_ID || ' ' || LAST_NAME || ' ' || EMP_AD);

END LOOP;

CLOSE c_EMP_;

END;
```

DECLARE	EMP_ID EMPEMP_	ID%type;
EMP ID, LAST	Γ NAME,EMP AD;	EXIT W

123 SHAIK AP 143 BASHA TN 111 TEJ IN 189 REDDY FR

Statement processed. 0.01 seconds

Run By	IN_A816_PLSQL_S39
Parsing Schema	IN_A816_PLSQL_S39