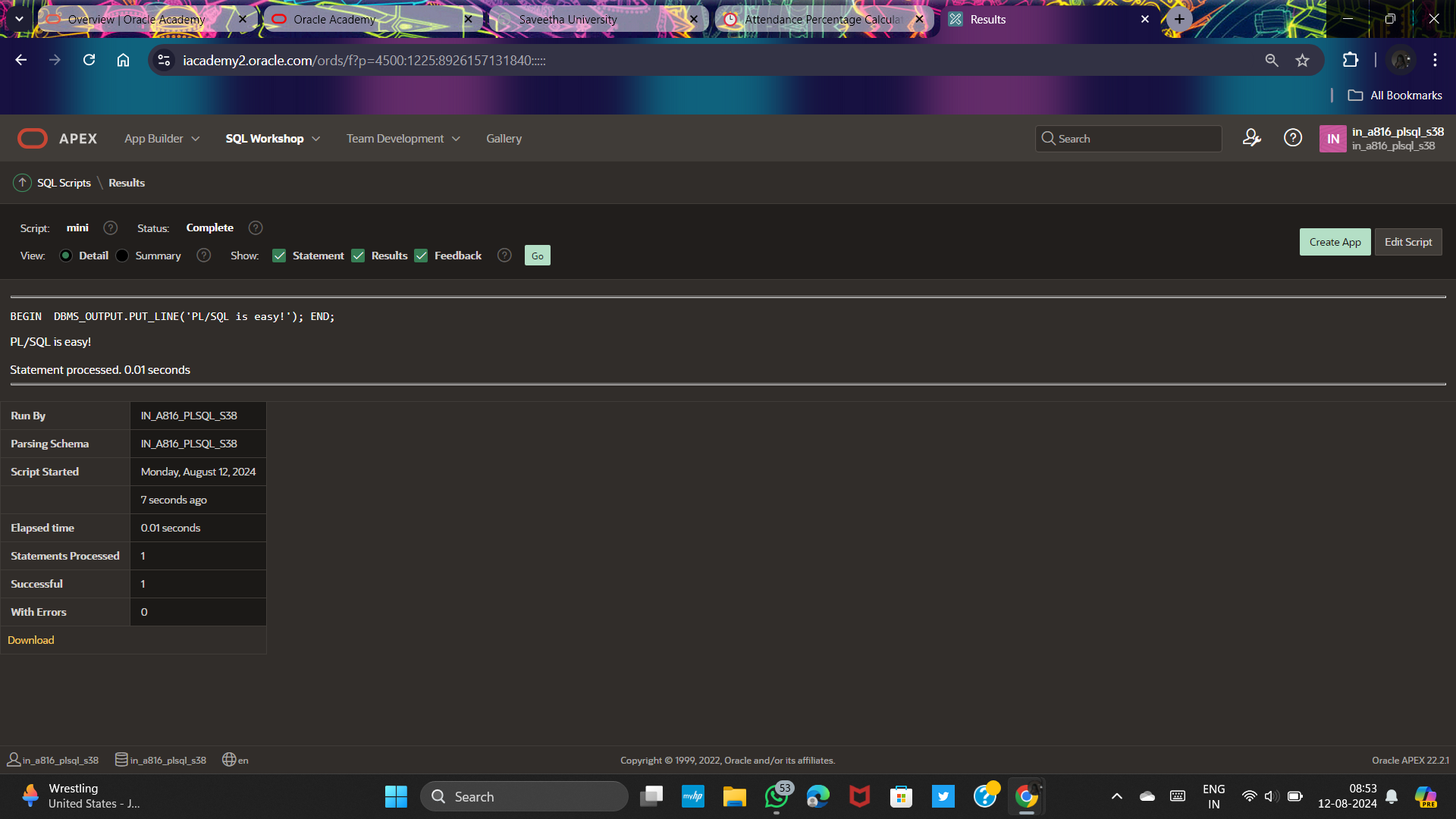
**EXAMPLE FOR ANONYMOUS BLOCKS:**

* **Executable section only(minimum required):**

BEGIN

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

END;

* **Declarative and executable sections:**

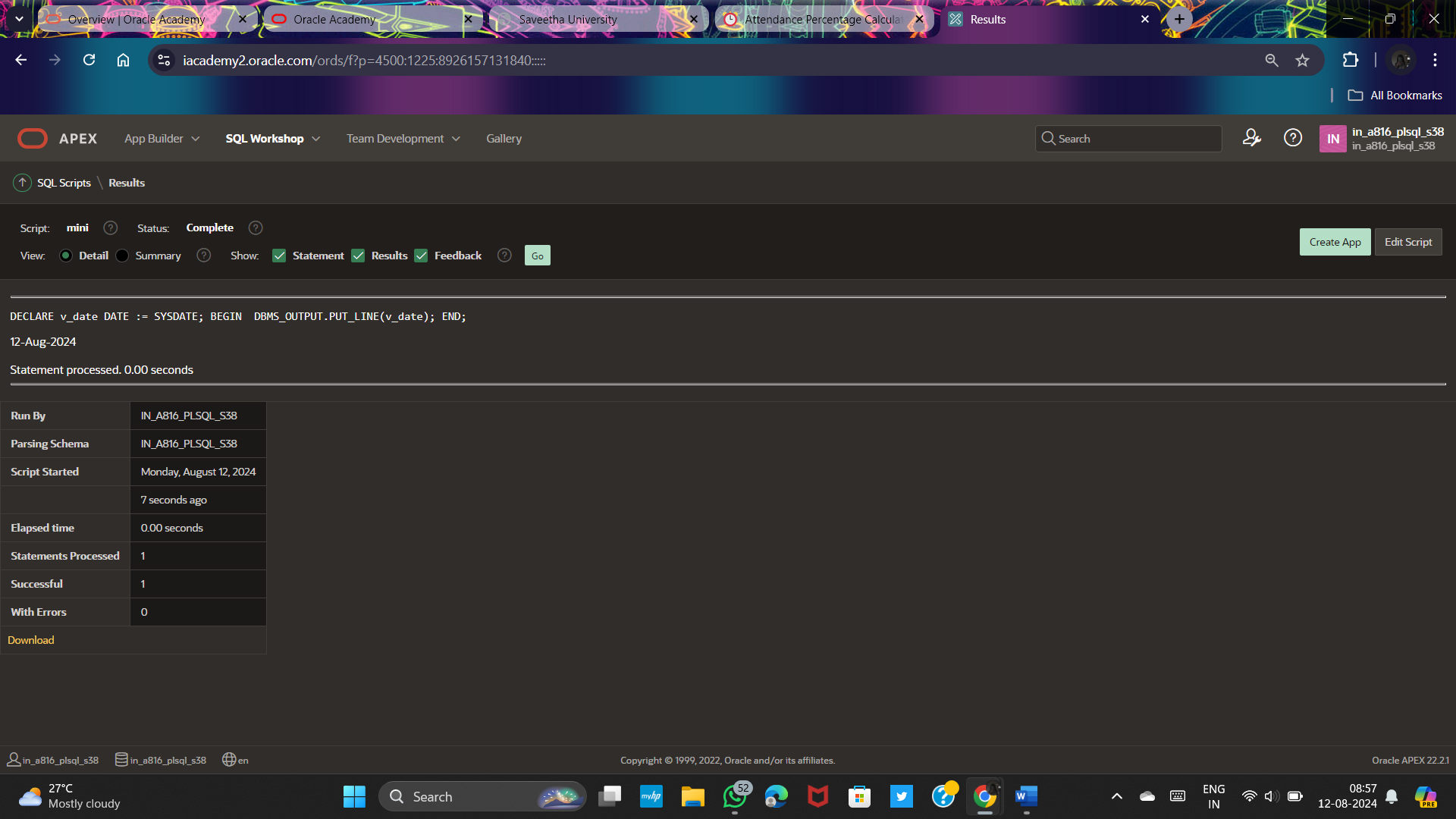
DECLARE

v\_date DATE := SYSDATE;

BEGIN

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

END;



**Example of anonymous blocks:**

* **Declarative,executable and exception sections:**

DECLARE

v\_firstname VARCHAR(25);

v\_lastname VARCHAR(25);

BEGIN

SELECT firstname,lastname

INTO v\_firstname,v\_lastname

FROM copyemp

WHERE lastname = 'govindaraju';

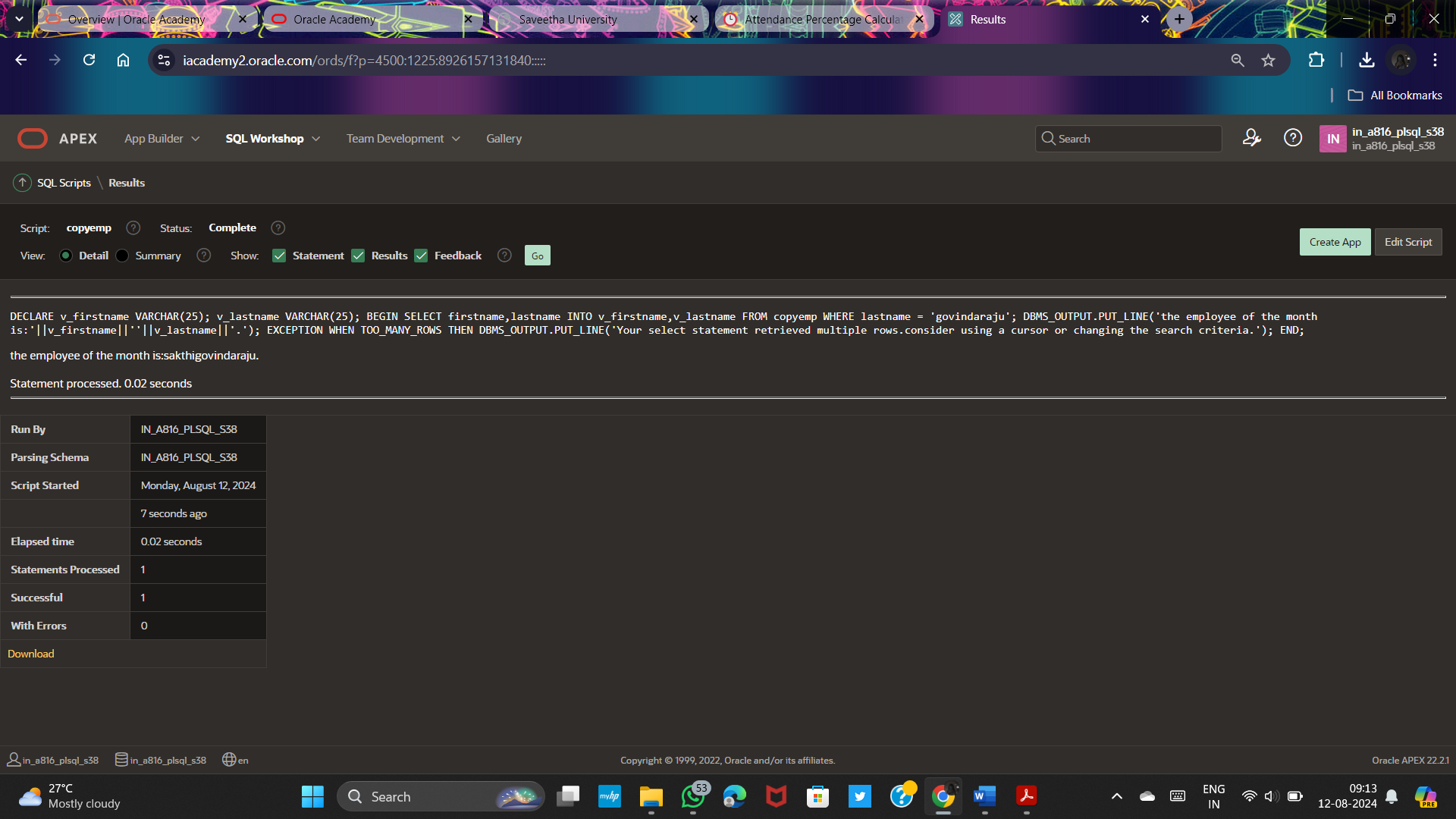
DBMS\_OUTPUT.PUT\_LINE('the employee of the month is:'||v\_firstname||''||v\_lastname||'.');

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;



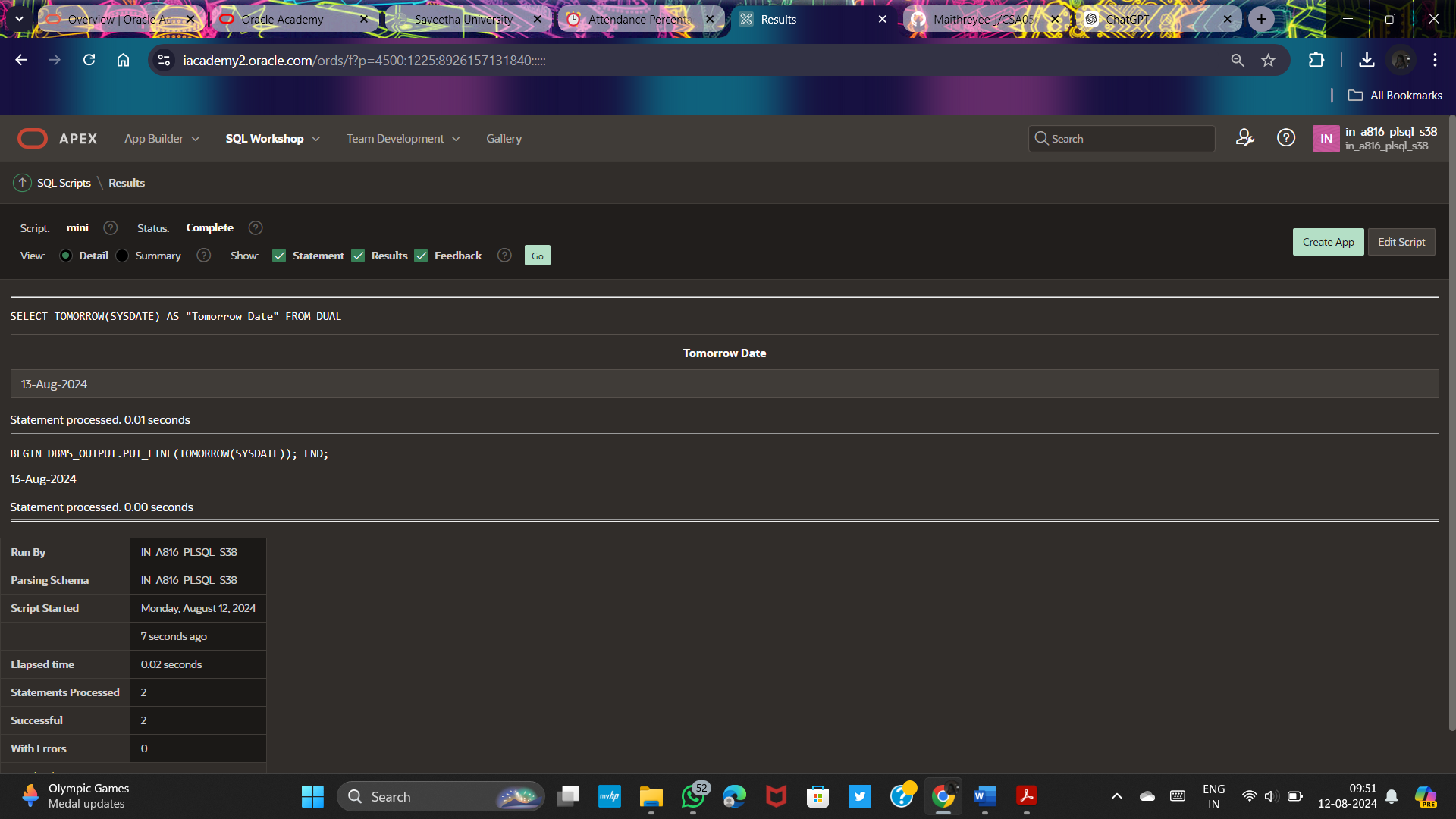
SELECT TOMORROW(SYSDATE) AS "Tomorrow Date"

FROM DUAL;

BEGIN

DBMS\_OUTPUT.PUT\_LINE(TOMORROW(SYSDATE));

END;



**Sum of N natural numbers:**

DECLARE

a integer := 10;

b integer := 20;

c integer;

f real;

BEGIN

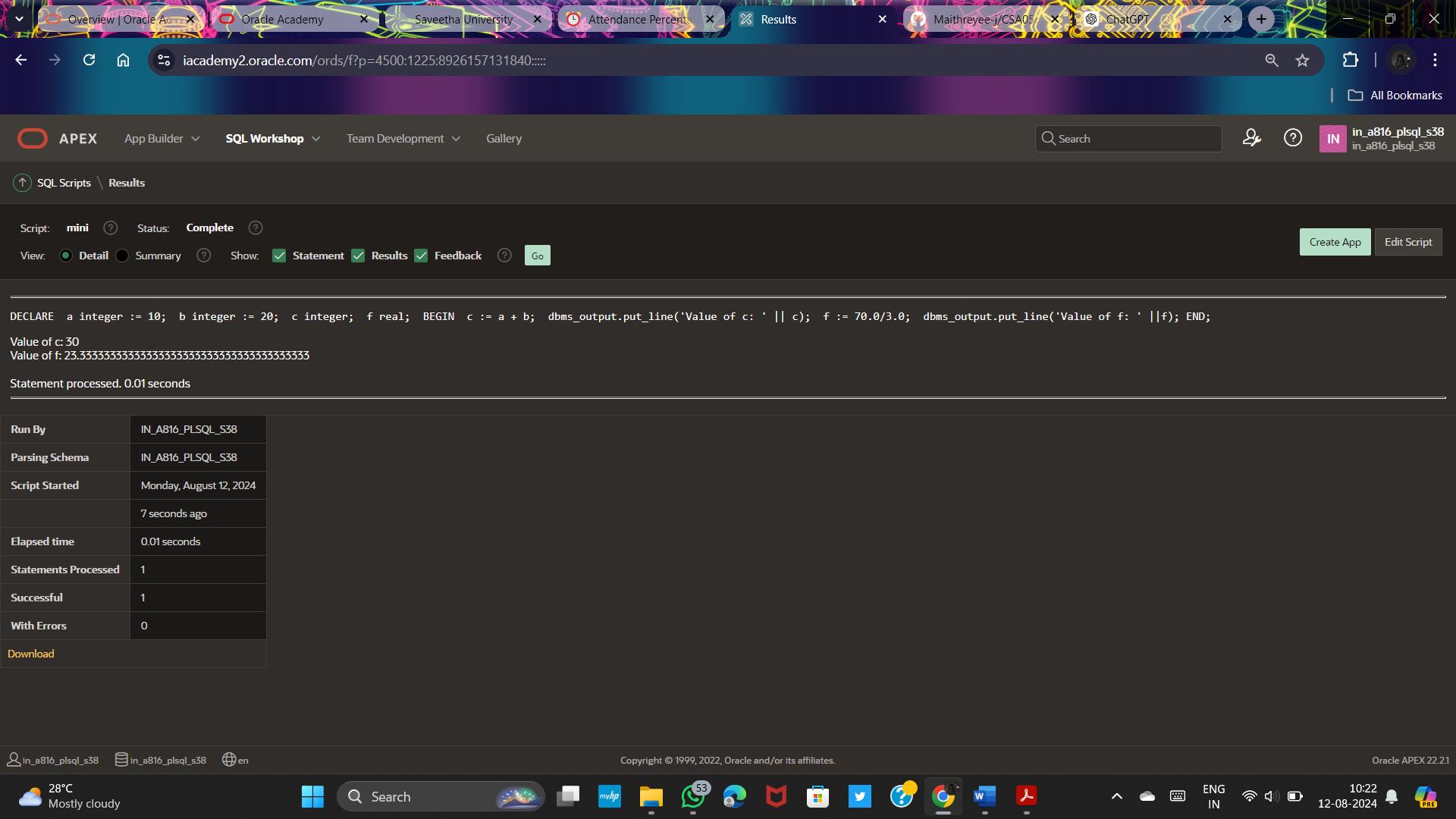
c := a + b;

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

f := 70.0/3.0;

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

END;



**Area and circumference:**

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

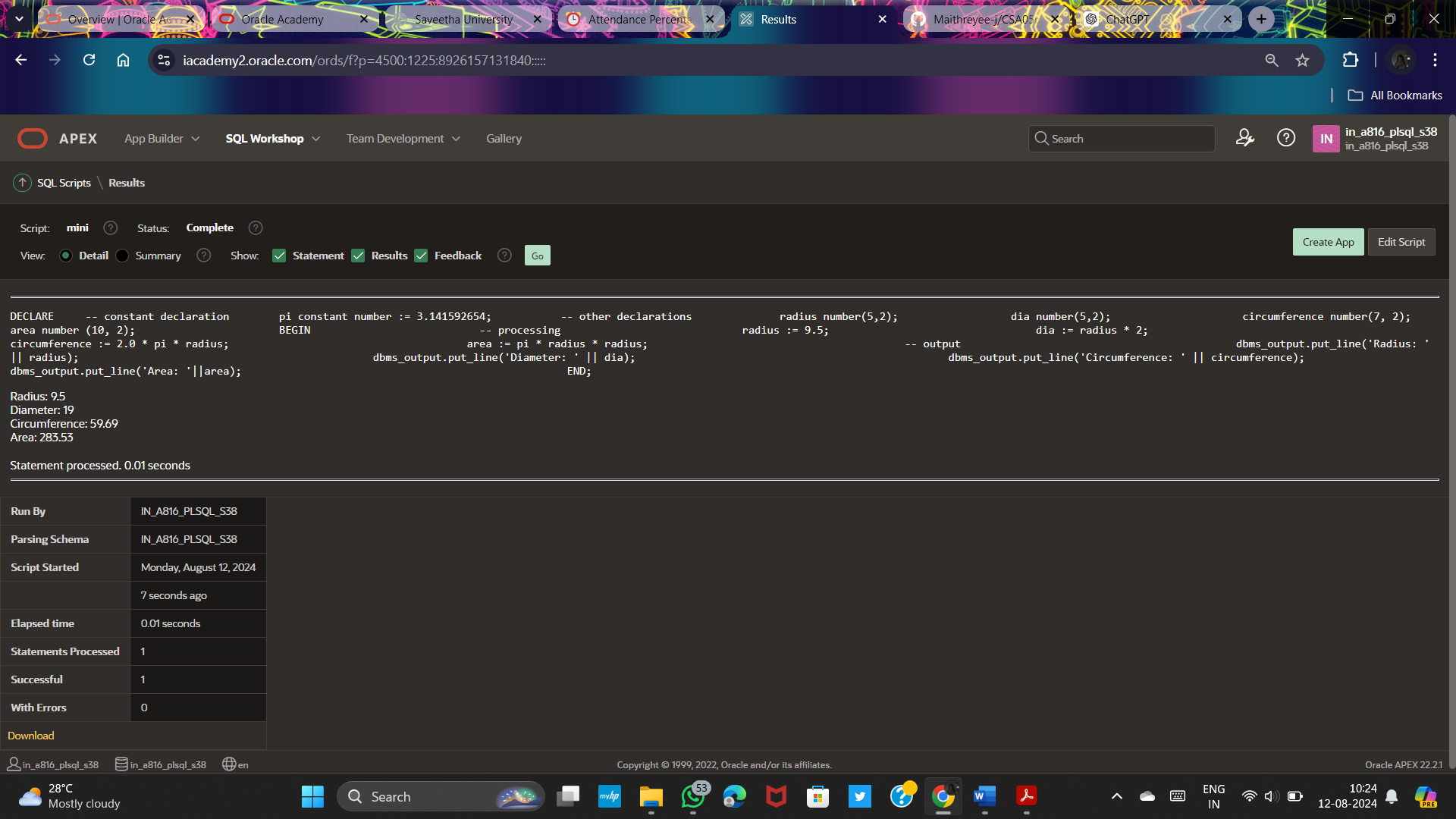
dbms\_output.put\_line('Radius: ' || radius);

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

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

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

END;



**Count of words and characters:(for loop)**

DECLARE

str VARCHAR2(40) := 'Tutorials Point';

nchars NUMBER(4) := 0;

nwords NUMBER(4) := 1;

s CHAR;

BEGIN

FOR i IN 1..Length(str) LOOP

s := Substr(str, i, 1);

nchars:= nchars+ 1;

IF s = ' ' THEN

nwords := nwords + 1;

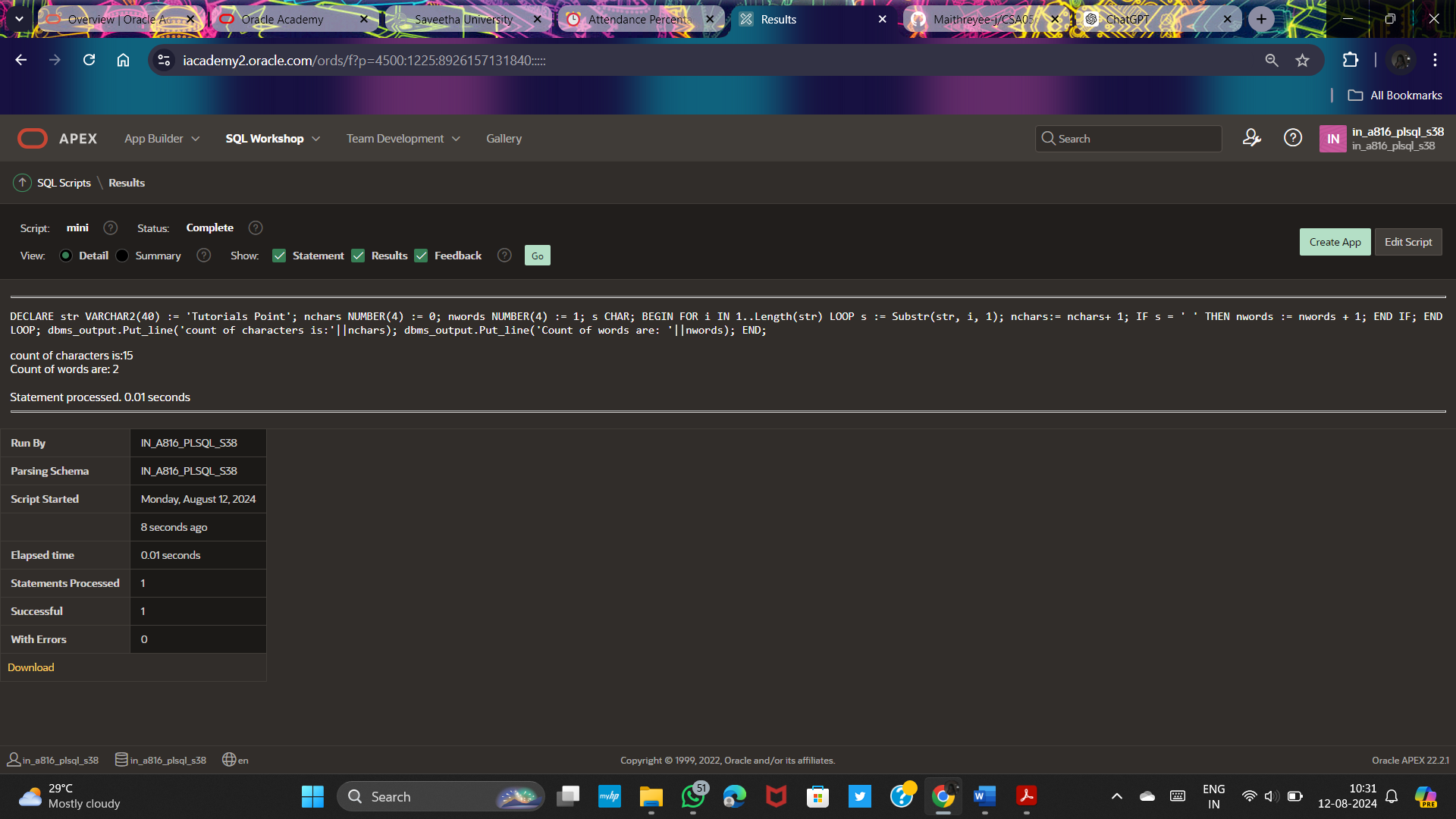
END IF;

END LOOP;

dbms\_output.Put\_line('count of characters is:'||nchars);

dbms\_output.Put\_line('Count of words are: '||nwords);

END;



**Using DBMS\_OUTPUT.PUT\_LINE Example:**

DECLARE

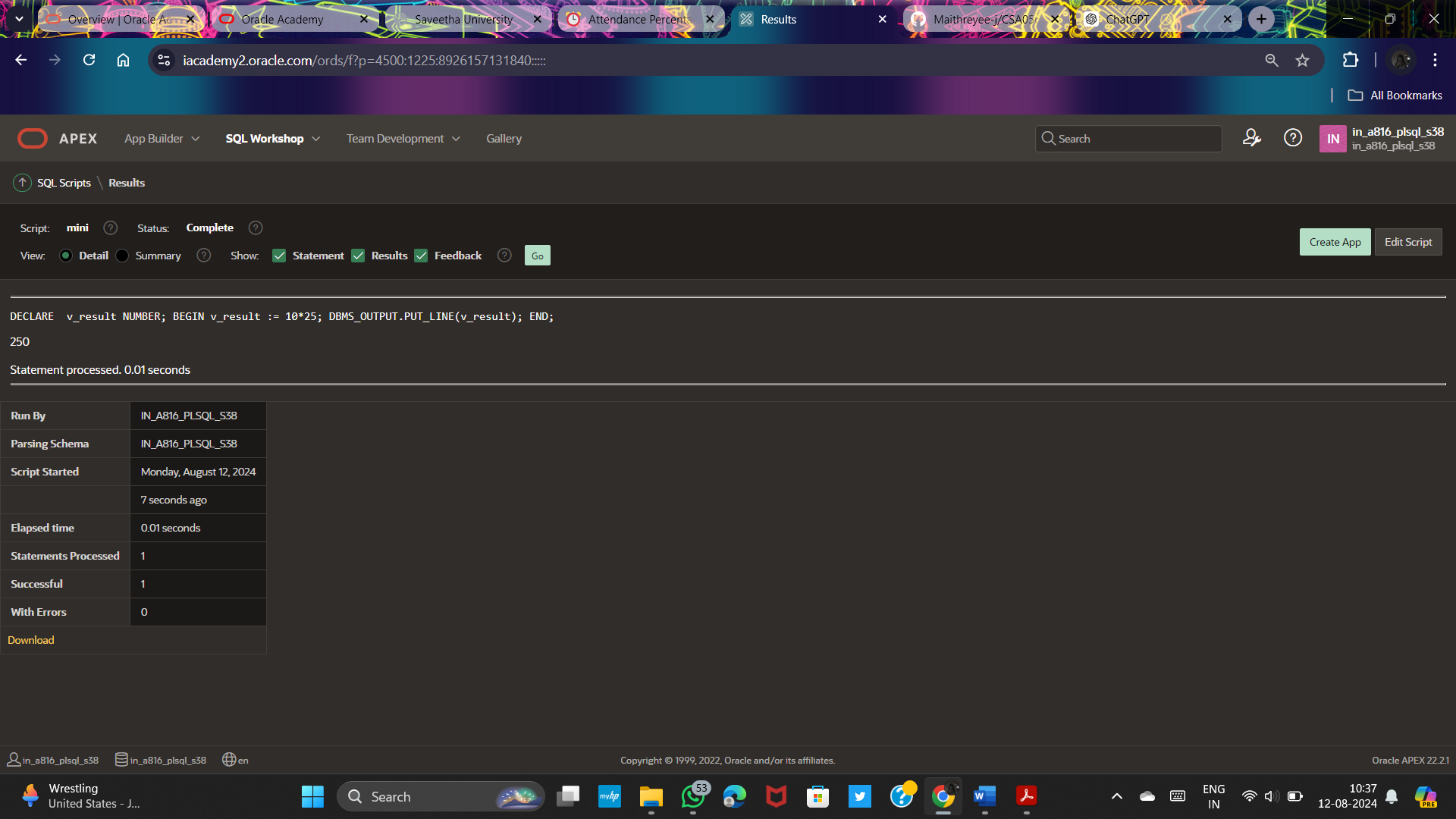
v\_result NUMBER;

BEGIN

v\_result := 10\*25;

DBMS\_OUTPUT.PUT\_LINE(v\_result);

END;



**Array:**

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');

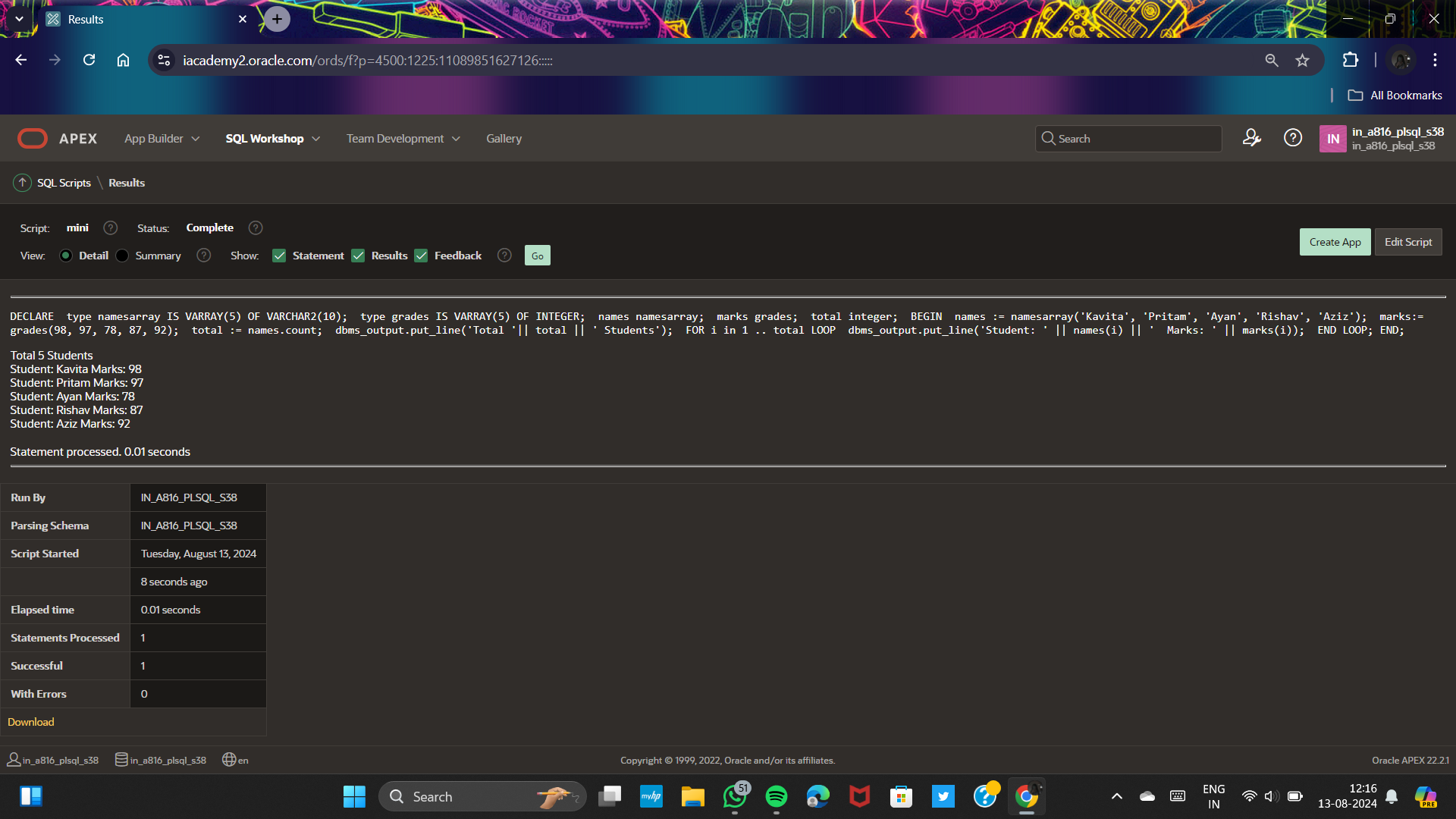
FOR i in 1 .. total LOOP

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

Marks: ' || marks(i));

 END LOOP;

END;



**Add FOR loop condition:**

DECLARE

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

type grades IS VARRAY(5) OF INTEGER;

type grade\_labels IS VARRAY(5) OF VARCHAR2(2);

names namesarray;

marks grades;

total integer;

grade\_label varchar(2);

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

IF marks(i) >= 90 THEN

grade\_label := 'A';

ELSIF marks(i) >= 80 THEN

grade\_label := 'B';

ELSIF marks(i) >= 70 THEN

grade\_label := 'C';

ELSIF marks(i) >= 60 THEN

grade\_label := 'D';

ELSE

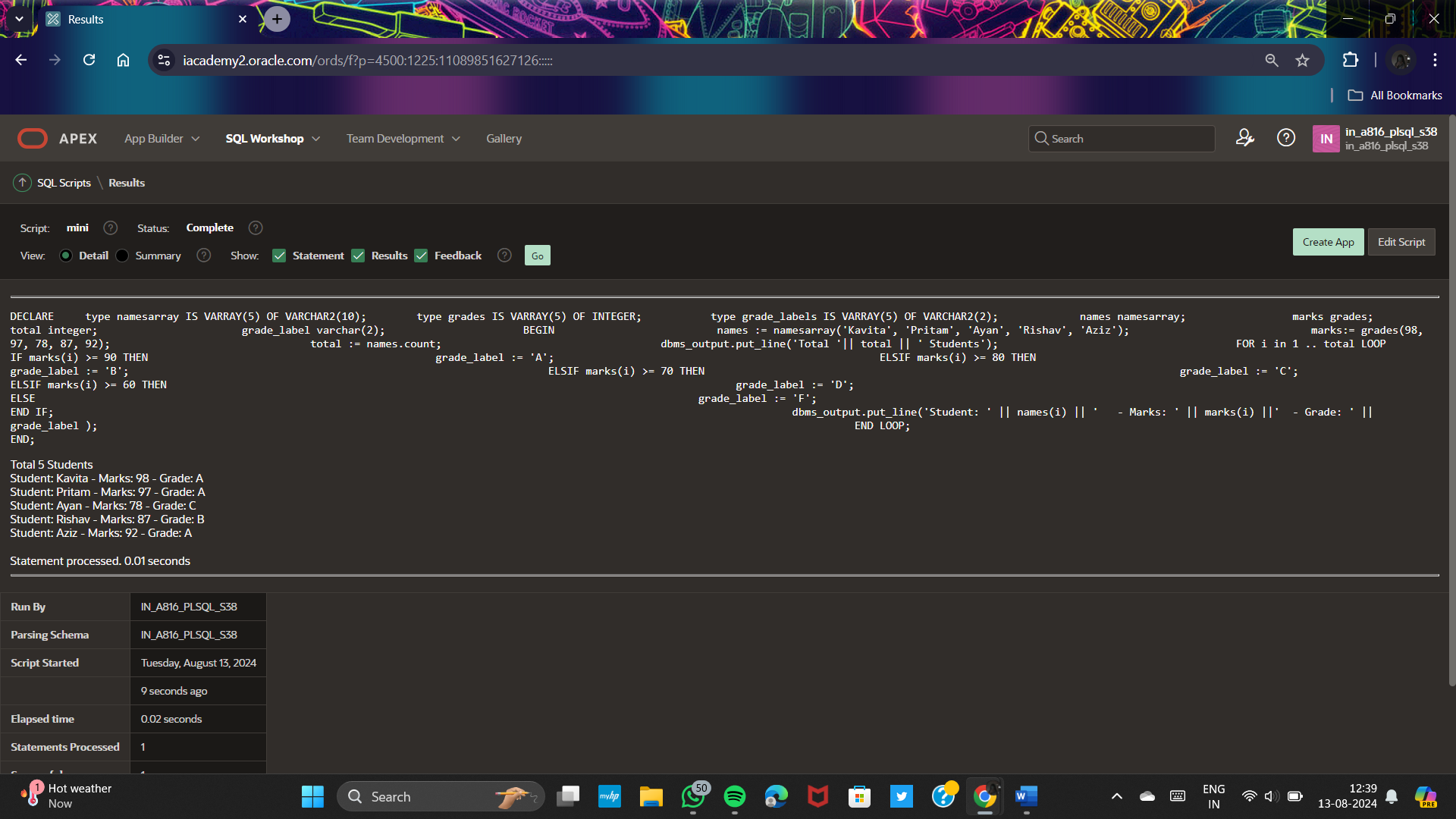
grade\_label := 'F';

END IF;

dbms\_output.put\_line('Student: ' || names(i) || ' - Marks: ' || marks(i) ||' - Grade: ' || grade\_label );

END LOOP;

END;



**Number of even and odd count:**

DECLARE

NUMBER\_TO\_CHECK NUMBER := 23146579;

ODD\_COUNT NUMBER := 0;

EVEN\_COUNT NUMBER := 0;

CURRENT\_DIGIT NUMBER;

BEGIN

WHILE NUMBER\_TO\_CHECK > 0 LOOP

CURRENT\_DIGIT := MOD(NUMBER\_TO\_CHECK, 10);

IF MOD(CURRENT\_DIGIT, 2) = 0 THEN

EVEN\_COUNT := EVEN\_COUNT + 1;

ELSE

ODD\_COUNT := ODD\_COUNT + 1;

END IF;

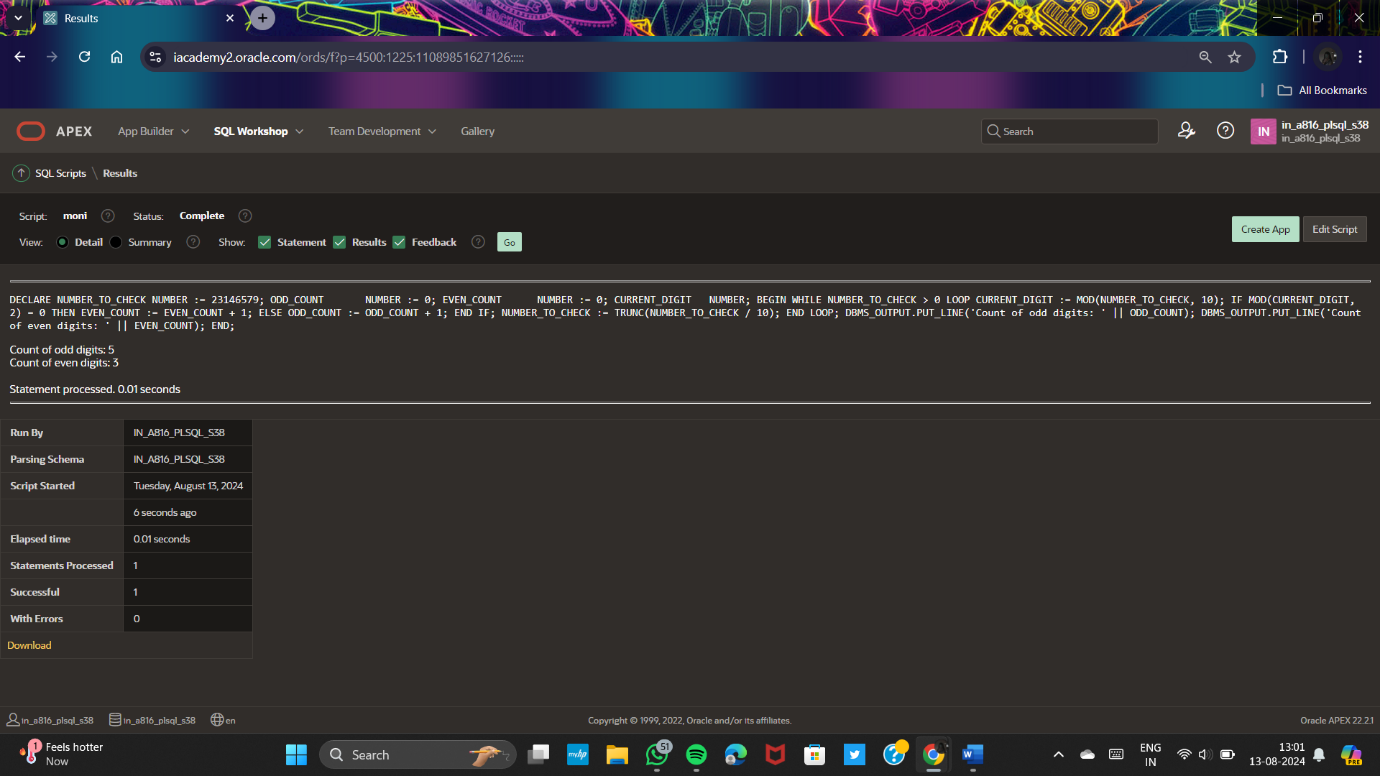
NUMBER\_TO\_CHECK := TRUNC(NUMBER\_TO\_CHECK / 10);

END LOOP;

DBMS\_OUTPUT.PUT\_LINE('Count of odd digits: ' || ODD\_COUNT);

DBMS\_OUTPUT.PUT\_LINE('Count of even digits: ' || EVEN\_COUNT);

END;



**Find the Minimum:**

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;

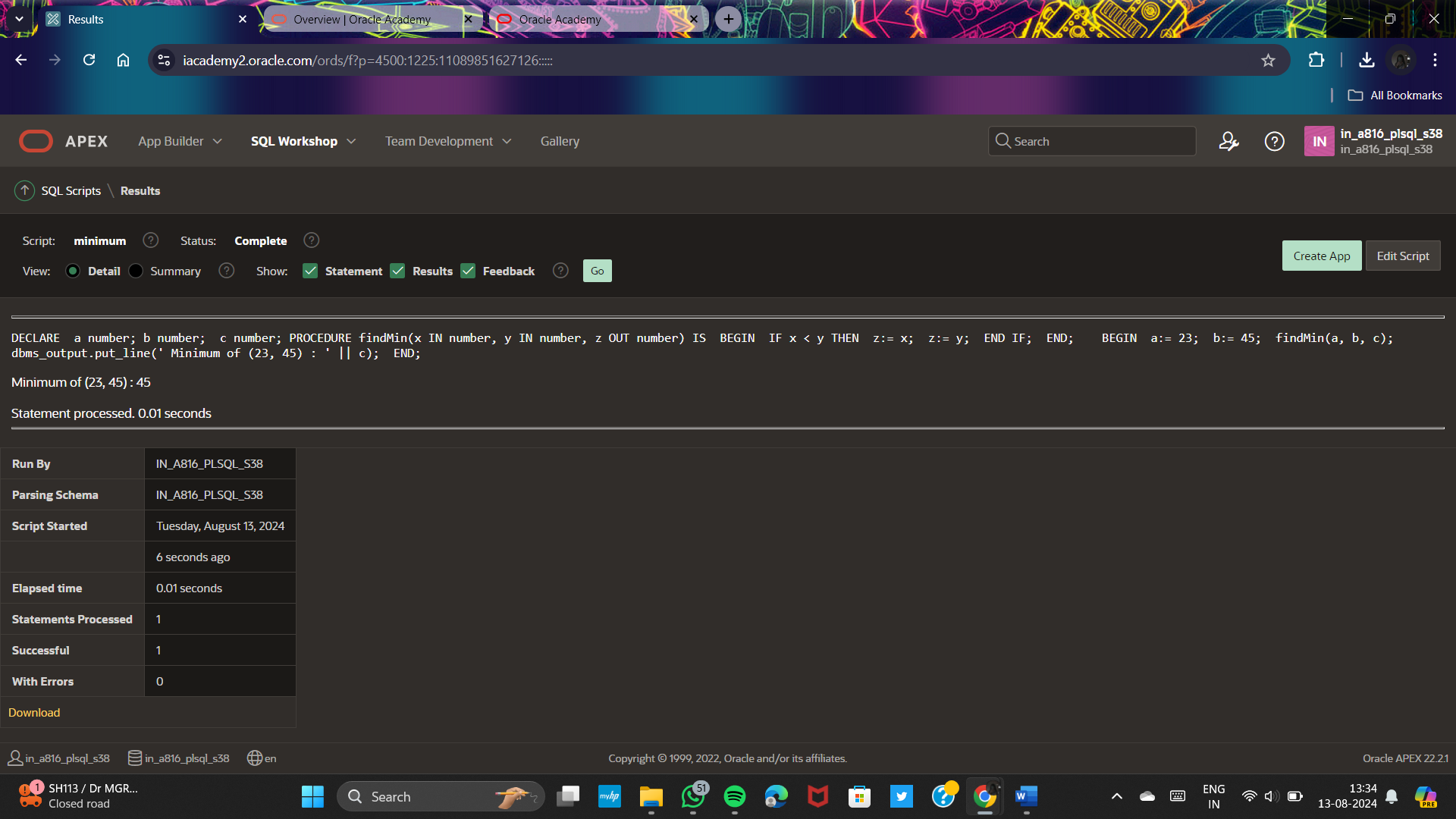
BEGIN

a:= 23;

b:= 45;

findMin(a, b, c);

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

END; 

**CALCULATOR:**

DECLARE

a NUMBER;

b NUMBER;

c NUMBER;

PROCEDURE Addtwo(x IN NUMBER, y IN NUMBER, z OUT NUMBER) IS

BEGIN

z := x + y;

END;

PROCEDURE Subtwo(x IN NUMBER, y IN NUMBER, z OUT NUMBER) IS

BEGIN

z := x - y;

END;

PROCEDURE Multwo(x IN NUMBER, y IN NUMBER, z OUT NUMBER) IS

BEGIN

z := x \* y;

END;

PROCEDURE Divtwo(x IN NUMBER, y IN NUMBER, z OUT NUMBER) IS

BEGIN

IF y = 0 THEN

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

RETURN;

ELSE

z := TRUNC(x / y);

END IF;

END;

PROCEDURE Modtwp(x IN NUMBER, y IN NUMBER, z OUT NUMBER) IS

BEGIN

z := MOD(x, y);

END;

BEGIN

a := 36;

b := 14;

Addtwo(a, b, c);

DBMS\_OUTPUT.PUT\_LINE('Addition of (36, 14): ' || c);

Subtwo(a, b, c);

DBMS\_OUTPUT.PUT\_LINE('Difference between (36, 14): ' || c);

Multwo(a, b, c);

DBMS\_OUTPUT.PUT\_LINE('Product of (36, 14): ' || c);

b := 12;

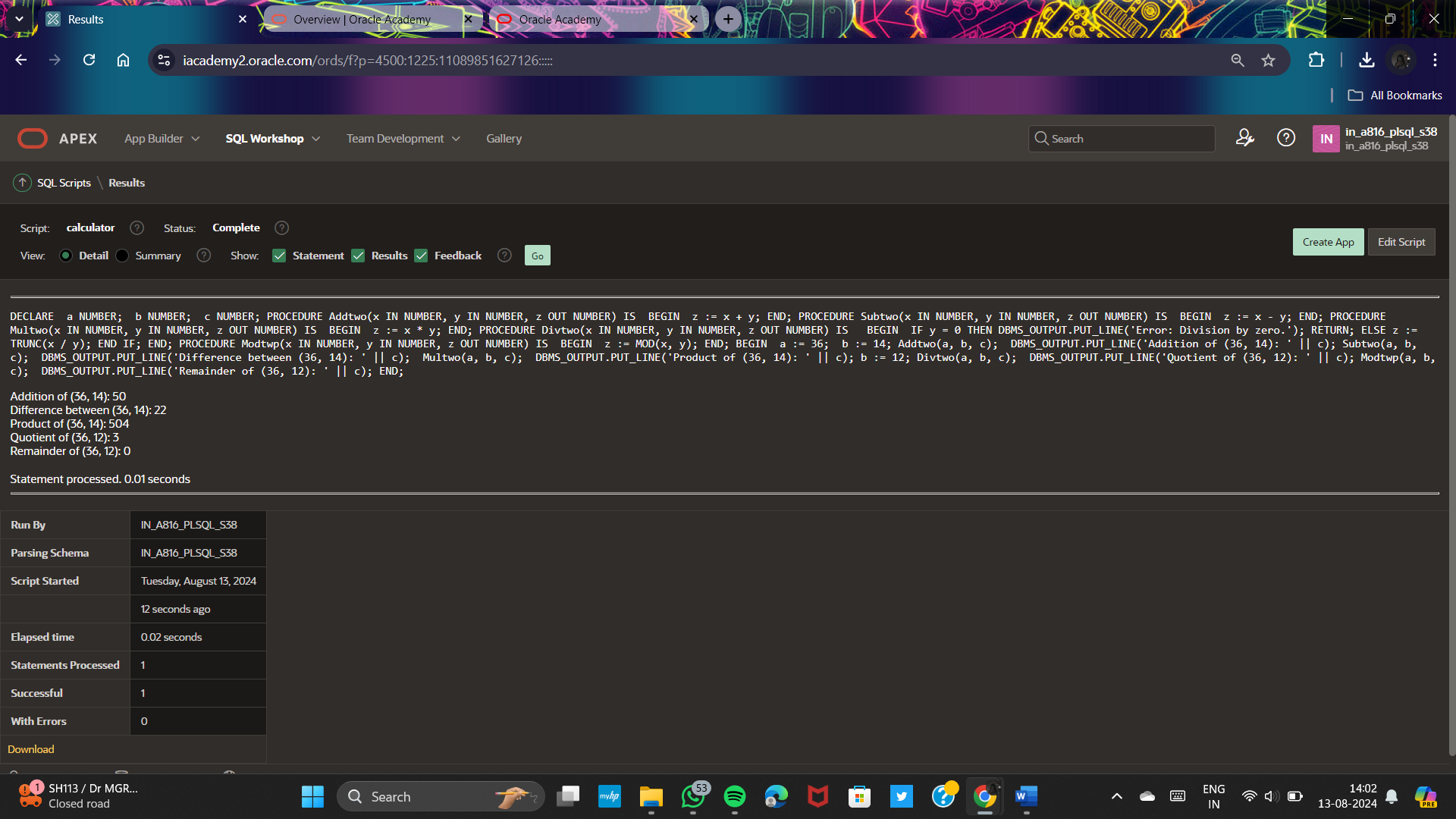
Divtwo(a, b, c);

DBMS\_OUTPUT.PUT\_LINE('Quotient of (36, 12): ' || c);

Modtwp(a, b, c);

DBMS\_OUTPUT.PUT\_LINE('Remainder of (36, 12): ' || c);

END;



**FACTORIAL OF A NUMBER (functions):**

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;

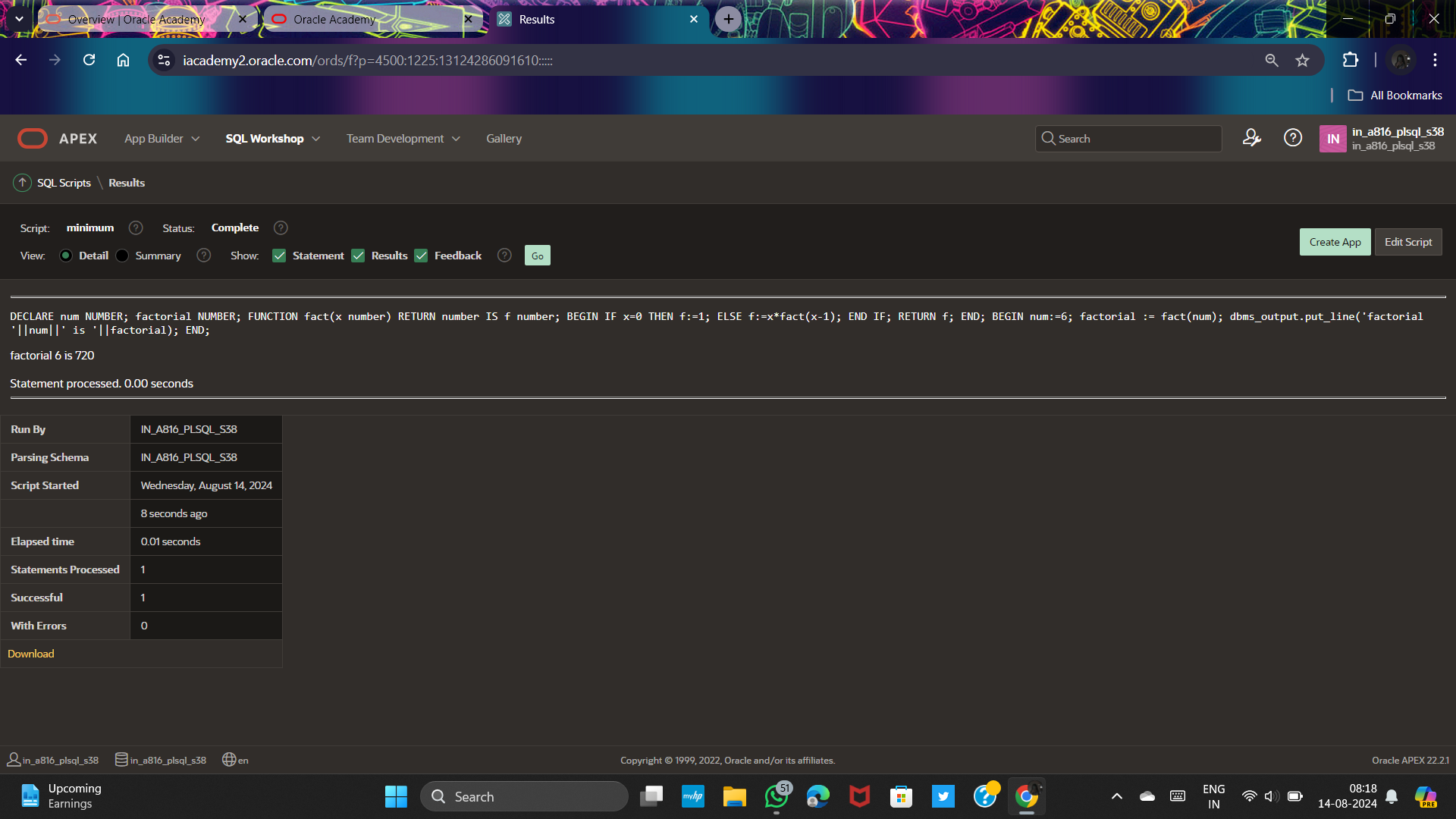
BEGIN

num:= 6;

factorial := fact(num);

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

END;



**FIND MAXIMUM :**

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

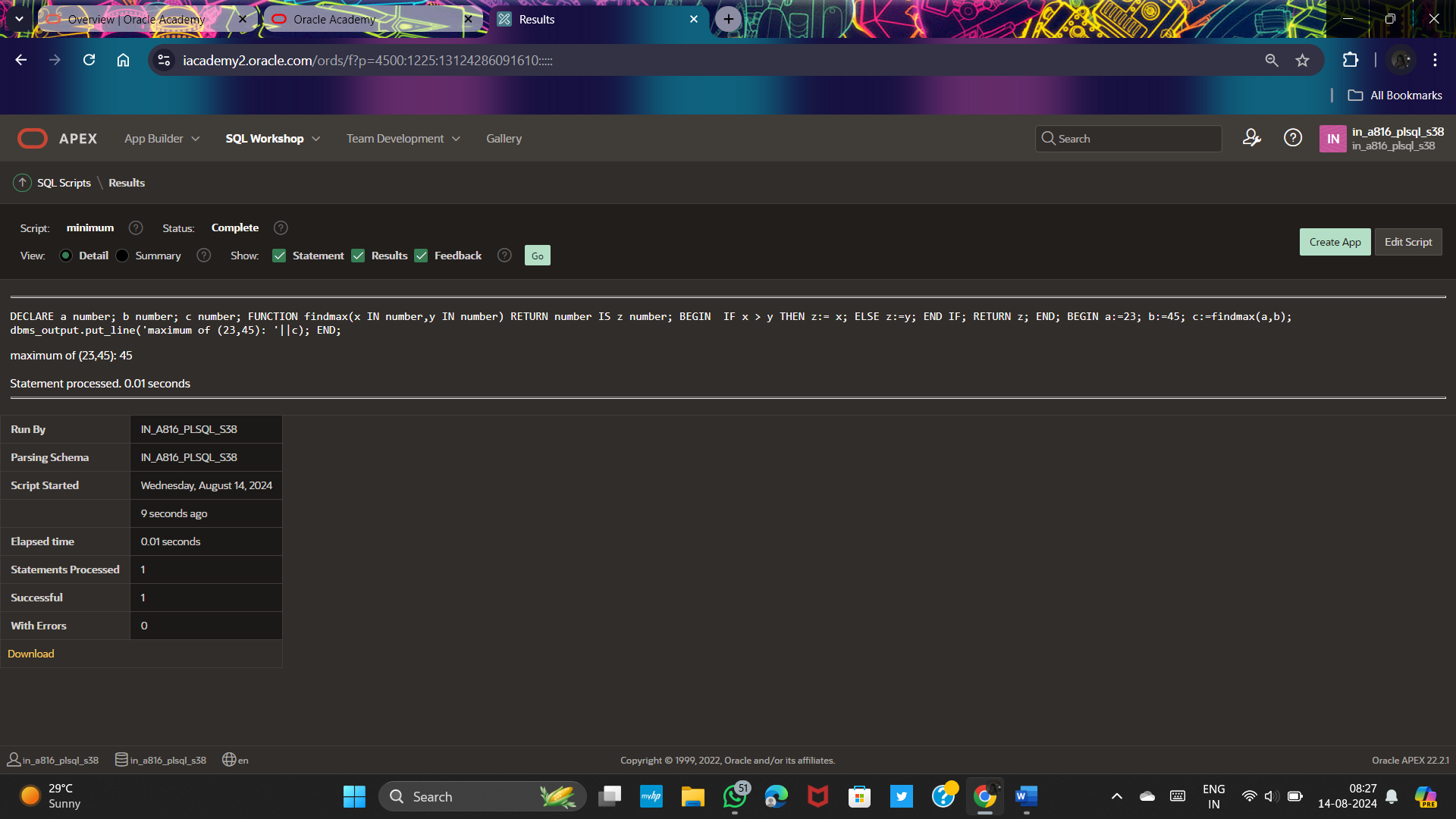
a:= 23;

b:= 45;

c := findMax(a, b);

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

END;



DECLARE

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;

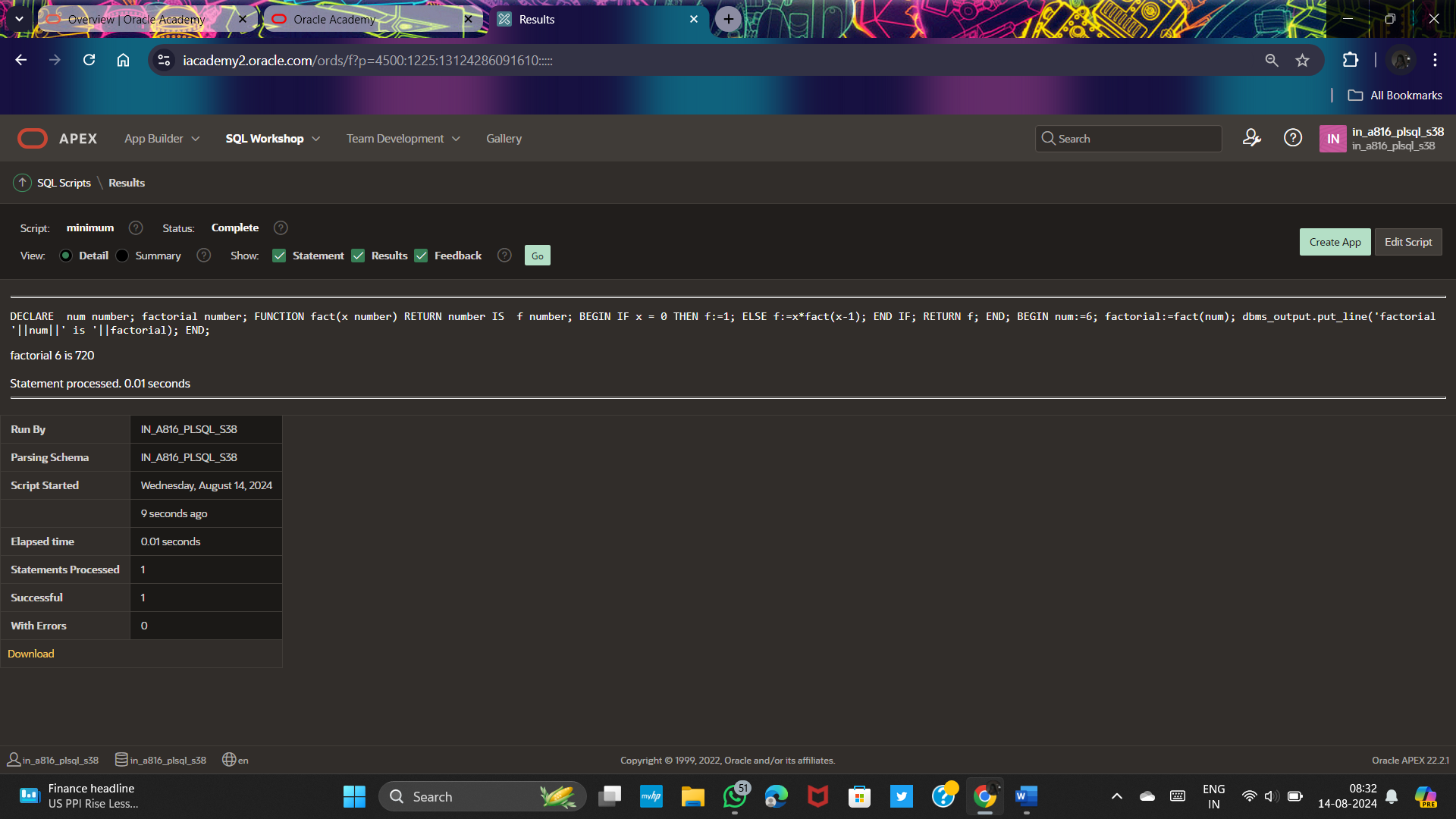
BEGIN

num:= 6;

factorial := fact(num);

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

END;



**FIBONACCI USING RECURSION:**

CREATE OR REPLACE FUNCTION fibonacci(n IN NUMBER) RETURN NUMBER IS

result NUMBER;

BEGIN

IF n <= 0 THEN

result := 0;

ELSIF n = 1 THEN

result := 1;

ELSE

result := fibonacci(n - 1) + fibonacci(n - 2);

END IF;

RETURN result;

END;

/

DECLARE

num\_terms NUMBER := 10;

i NUMBER;

fib\_num NUMBER;

BEGIN

FOR i IN 1..num\_terms LOOP

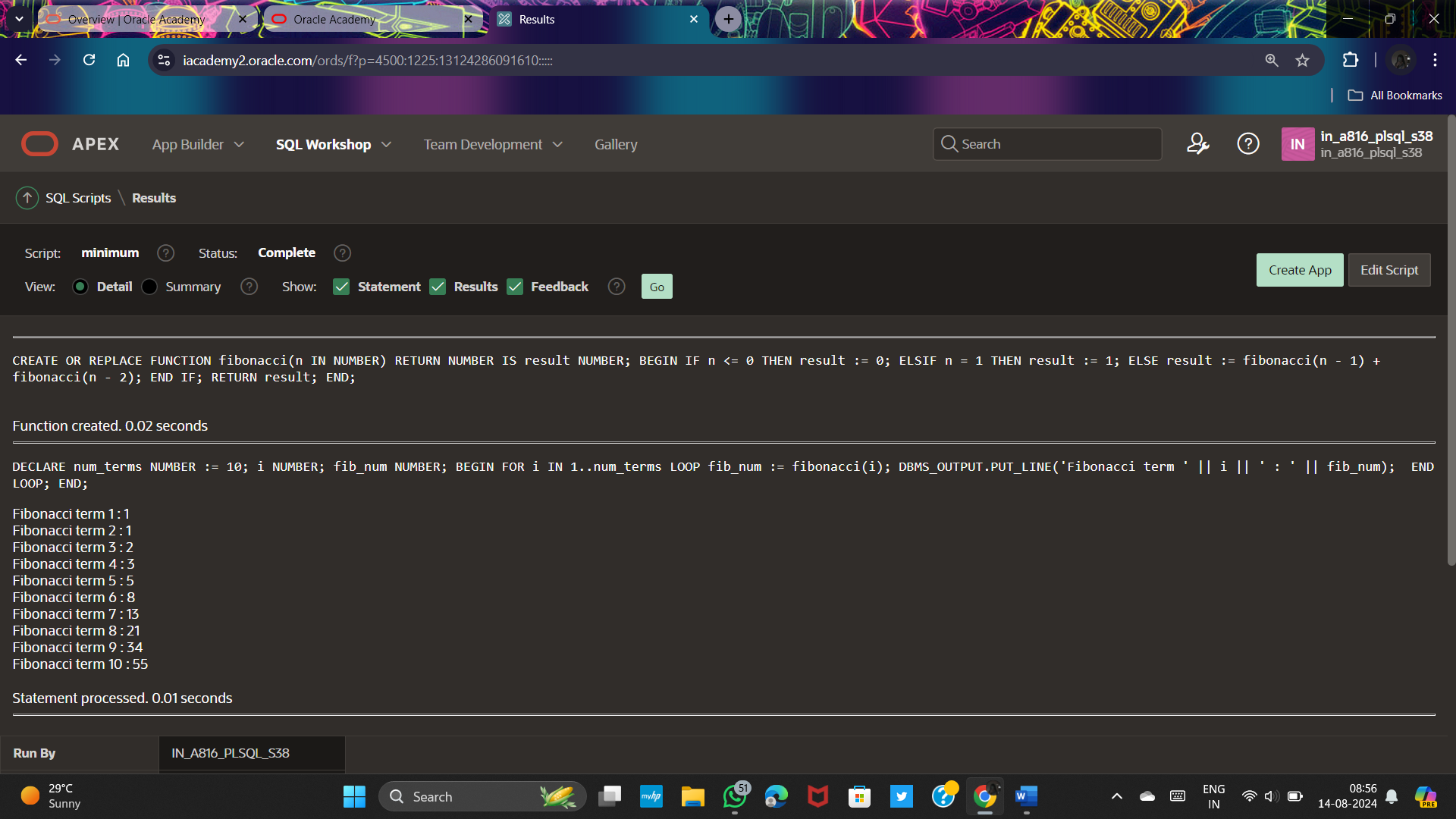
fib\_num := fibonacci(i);

DBMS\_OUTPUT.PUT\_LINE('Fibonacci term ' || i || ' : ' || fib\_num);

END LOOP;

END;

/



DECLARE

c\_emp\_id empl.EMPLOYEE\_ID%TYPE;

c\_fname empl.FIRSTNAME%TYPE;

c\_lname empl.LASTNAME%TYPE;

c\_deptid empl.DEPTID%TYPE;

CURSOR c\_employee IS

SELECT EMPLOYEE\_ID, FIRSTNAME, LASTNAME, DEPTID

FROM empl;

BEGIN

OPEN c\_employee;

LOOP

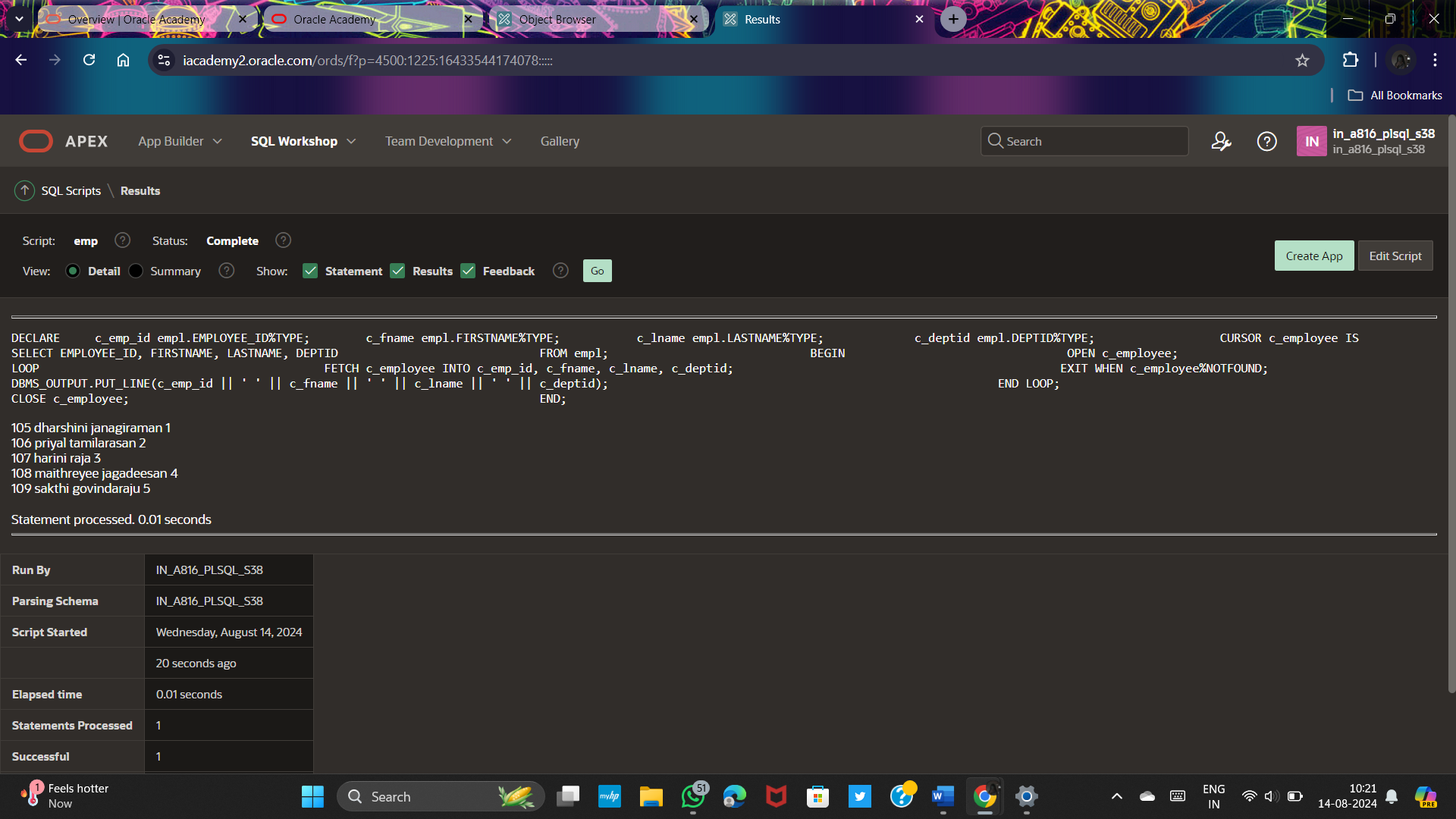
FETCH c\_employee INTO c\_emp\_id, c\_fname, c\_lname, c\_deptid;

EXIT WHEN c\_employee%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE(c\_emp\_id || ' ' || c\_fname || ' ' || c\_lname || ' ' || c\_deptid);

END LOOP;

CLOSE c\_employee;

END; 

DECLARE

c\_person\_id person.PERSON\_ID%TYPE;

c\_person\_name person.PERSON\_NAME%TYPE;

c\_email person.EMAIL%TYPE;

c\_phone\_number person.PHONE\_NUMBER%TYPE;

CURSOR c\_persons IS

SELECT PERSON\_ID, PERSON\_NAME, EMAIL, PHONE\_NUMBER

FROM person;

BEGIN

OPEN c\_persons;

LOOP

FETCH c\_persons INTO c\_person\_id, c\_person\_name, c\_email, c\_phone\_number;

EXIT WHEN c\_persons%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE(c\_person\_id || ' ' || c\_person\_name || ' ' || c\_email || ' ' || c\_phone\_number);

END LOOP;

CLOSE c\_persons;

END;

