

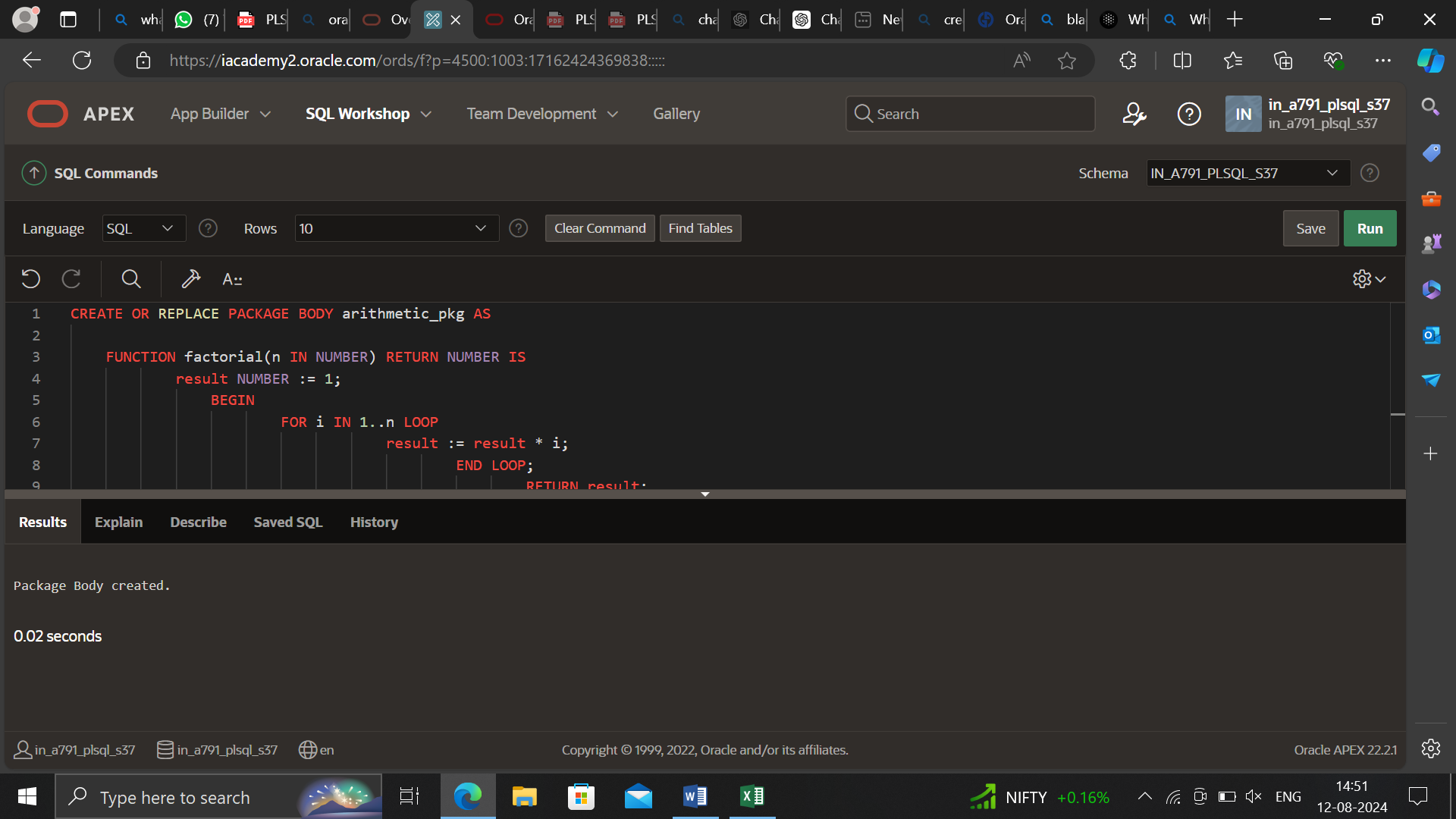
CREATE OR REPLACE PACKAGE arithmetic\_pkg AS

-- Public function to calculate factorial

FUNCTION factorial(n IN NUMBER) RETURN NUMBER;

END arithmetic\_pkg;

/



CREATE OR REPLACE PACKAGE BODY arithmetic\_pkg AS

FUNCTION factorial(n IN NUMBER) RETURN NUMBER IS

result NUMBER := 1;

BEGIN

FOR i IN 1..n LOOP

result := result \* i;

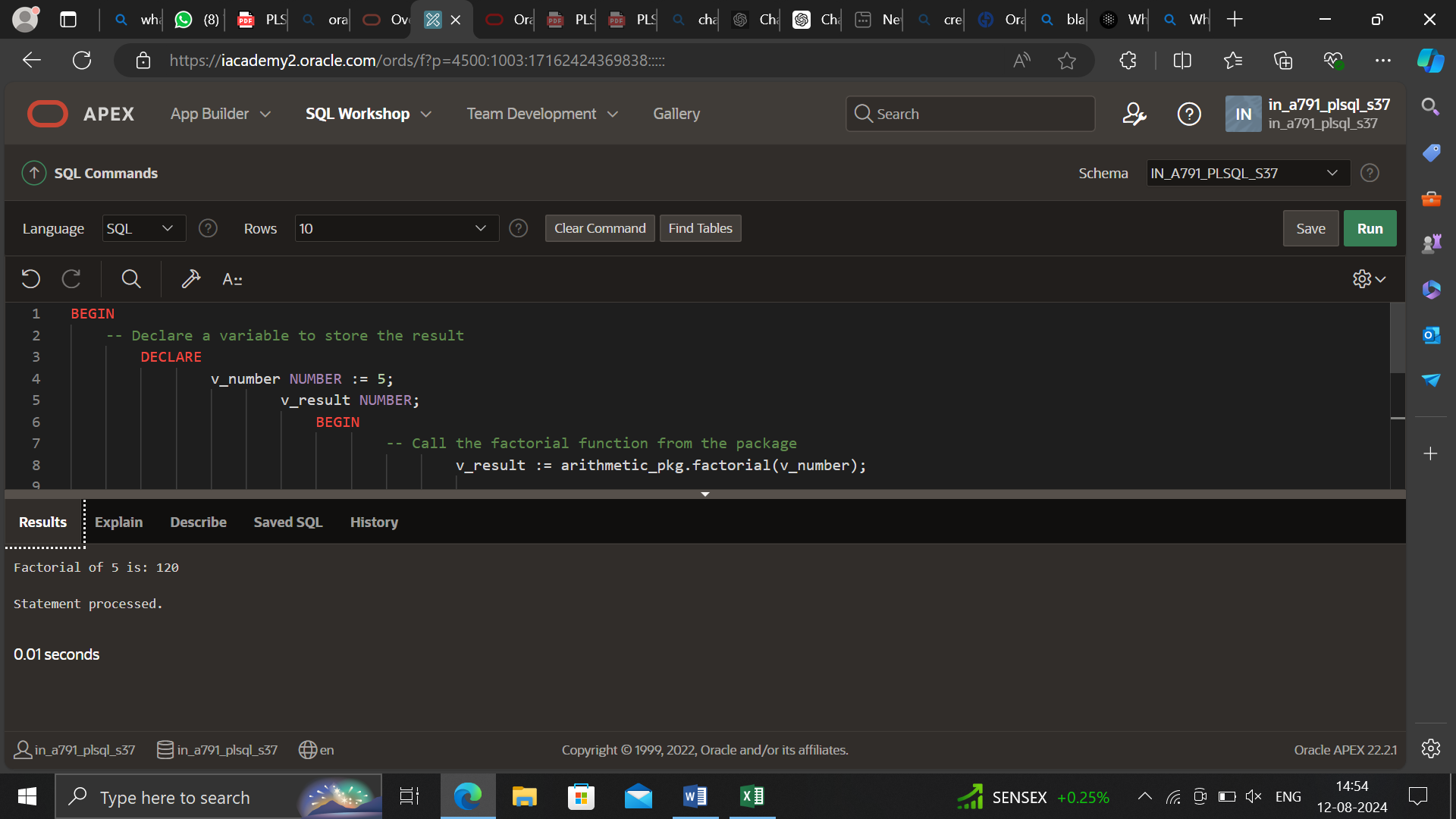
END LOOP;

RETURN result;

END factorial;

END arithmetic\_pkg;

/



BEGIN

-- Declare a variable to store the result

DECLARE

v\_number NUMBER := 5;

v\_result NUMBER;

BEGIN

-- Call the factorial function from the package

v\_result := arithmetic\_pkg.factorial(v\_number);

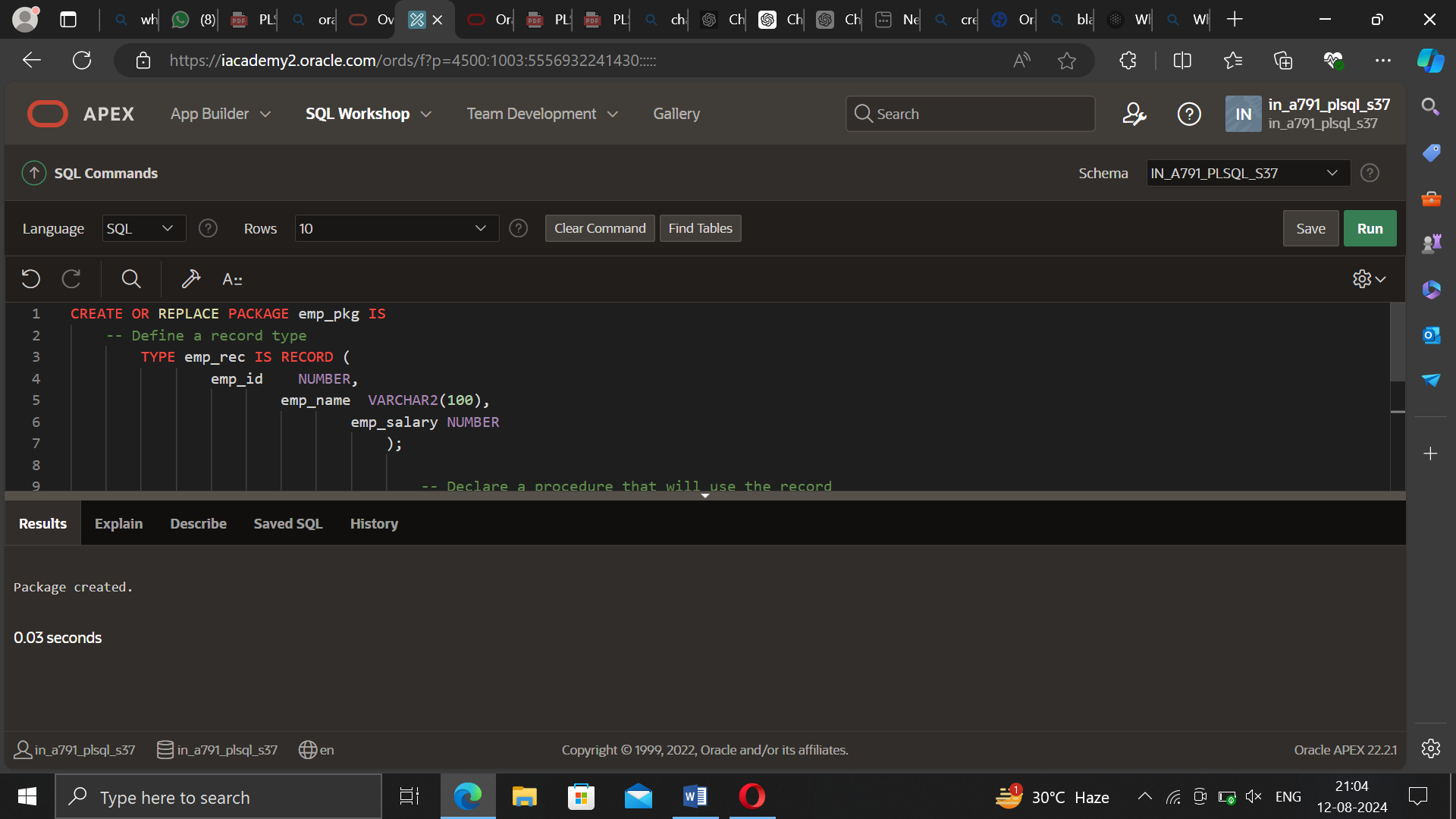
-- Print the result

DBMS\_OUTPUT.PUT\_LINE('Factorial of ' || v\_number || ' is: ' || v\_result);

END;

END;

/



CREATE OR REPLACE PACKAGE emp\_pkg IS

-- Define a record type

TYPE emp\_rec IS RECORD (

emp\_id NUMBER,

emp\_name VARCHAR2(100),

emp\_salary NUMBER

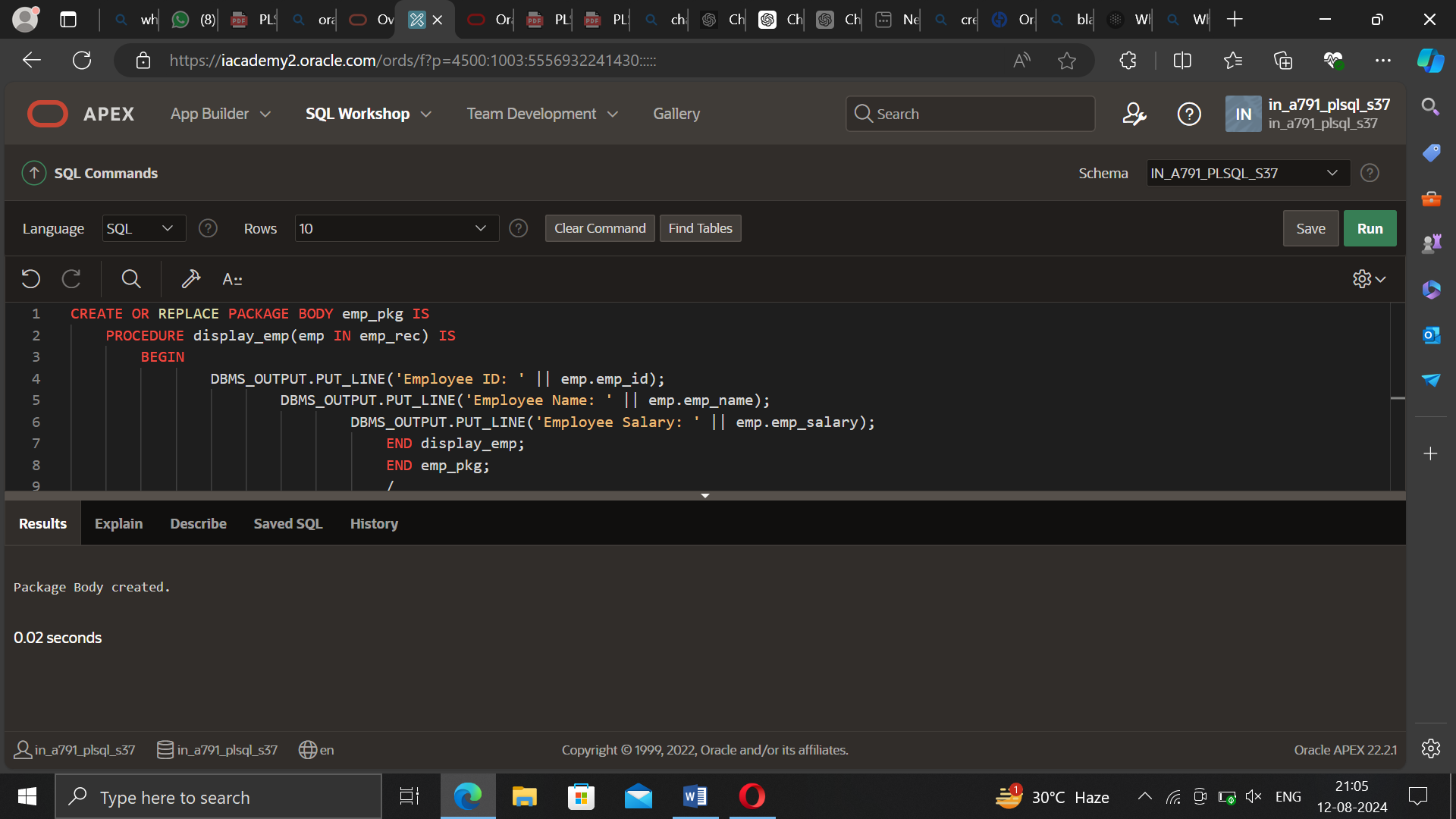
);

-- Declare a procedure that will use the record

PROCEDURE display\_emp(emp IN emp\_rec);

END emp\_pkg;

/



CREATE OR REPLACE PACKAGE BODY emp\_pkg IS

PROCEDURE display\_emp(emp IN emp\_rec) IS

BEGIN

DBMS\_OUTPUT.PUT\_LINE('Employee ID: ' || emp.emp\_id);

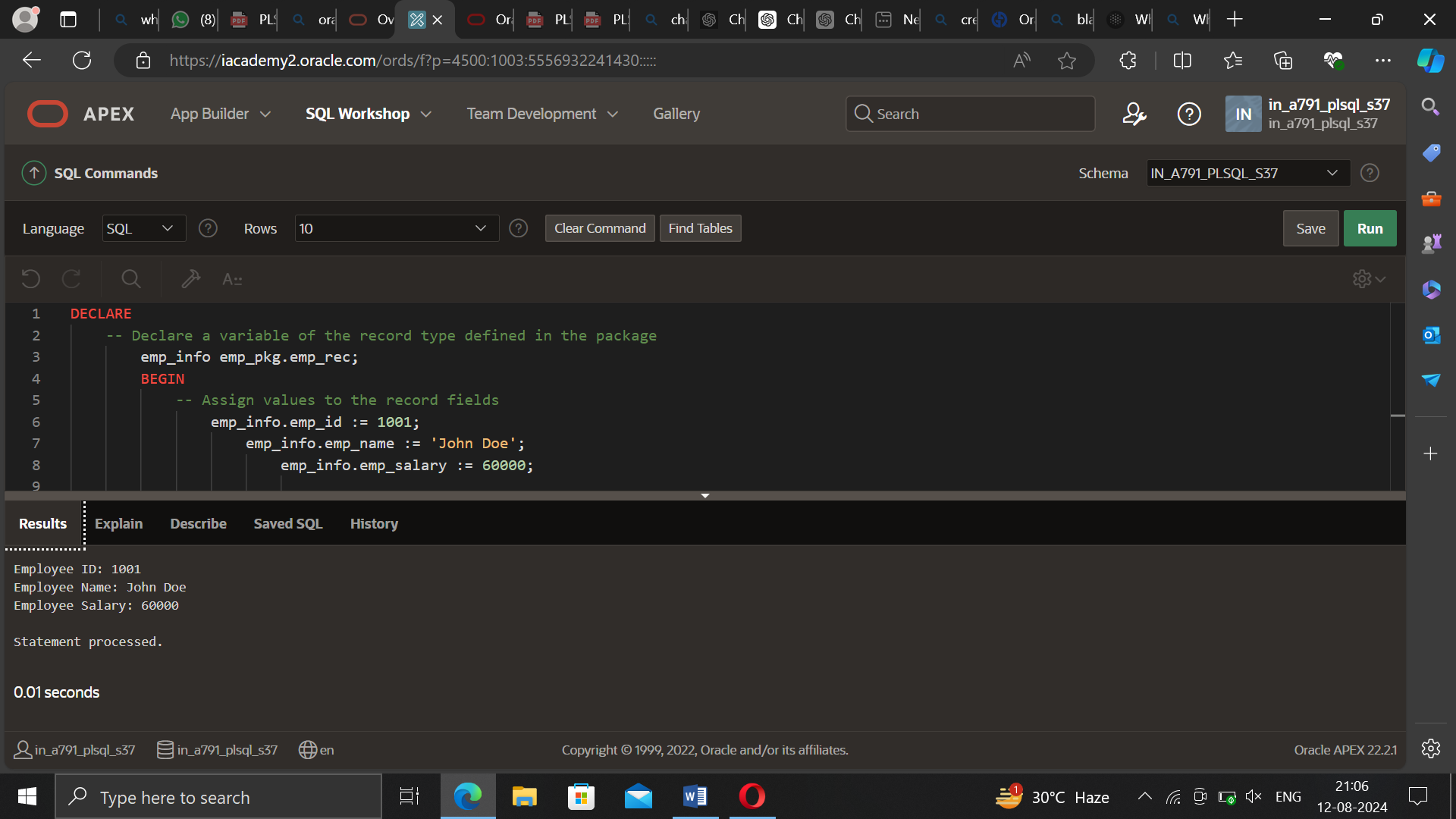
DBMS\_OUTPUT.PUT\_LINE('Employee Name: ' || emp.emp\_name);

DBMS\_OUTPUT.PUT\_LINE('Employee Salary: ' || emp.emp\_salary);

END display\_emp;

END emp\_pkg;

/



DECLARE

-- Declare a variable of the record type defined in the package

emp\_info emp\_pkg.emp\_rec;

BEGIN

-- Assign values to the record fields

emp\_info.emp\_id := 1001;

emp\_info.emp\_name := 'John Doe';

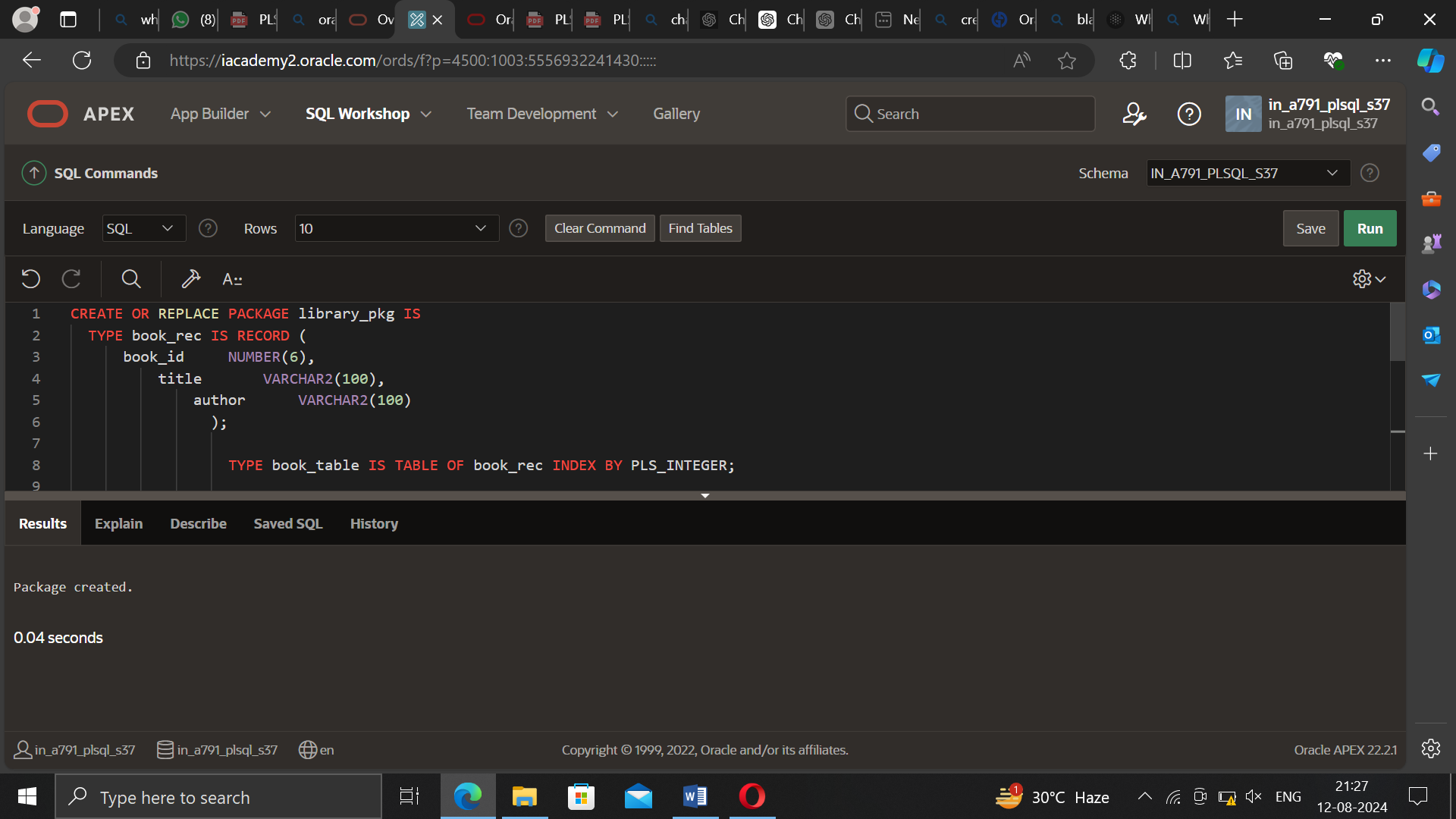
emp\_info.emp\_salary := 60000;

-- Call the procedure and pass the record as an argument

emp\_pkg.display\_emp(emp\_info);

END;

/



CREATE OR REPLACE PACKAGE library\_pkg IS

TYPE book\_rec IS RECORD (

book\_id NUMBER(6),

title VARCHAR2(100),

author VARCHAR2(100)

);

TYPE book\_table IS TABLE OF book\_rec INDEX BY PLS\_INTEGER;

PROCEDURE add\_book(

p\_books IN OUT book\_table,

p\_book\_id IN NUMBER,

p\_title IN VARCHAR2,

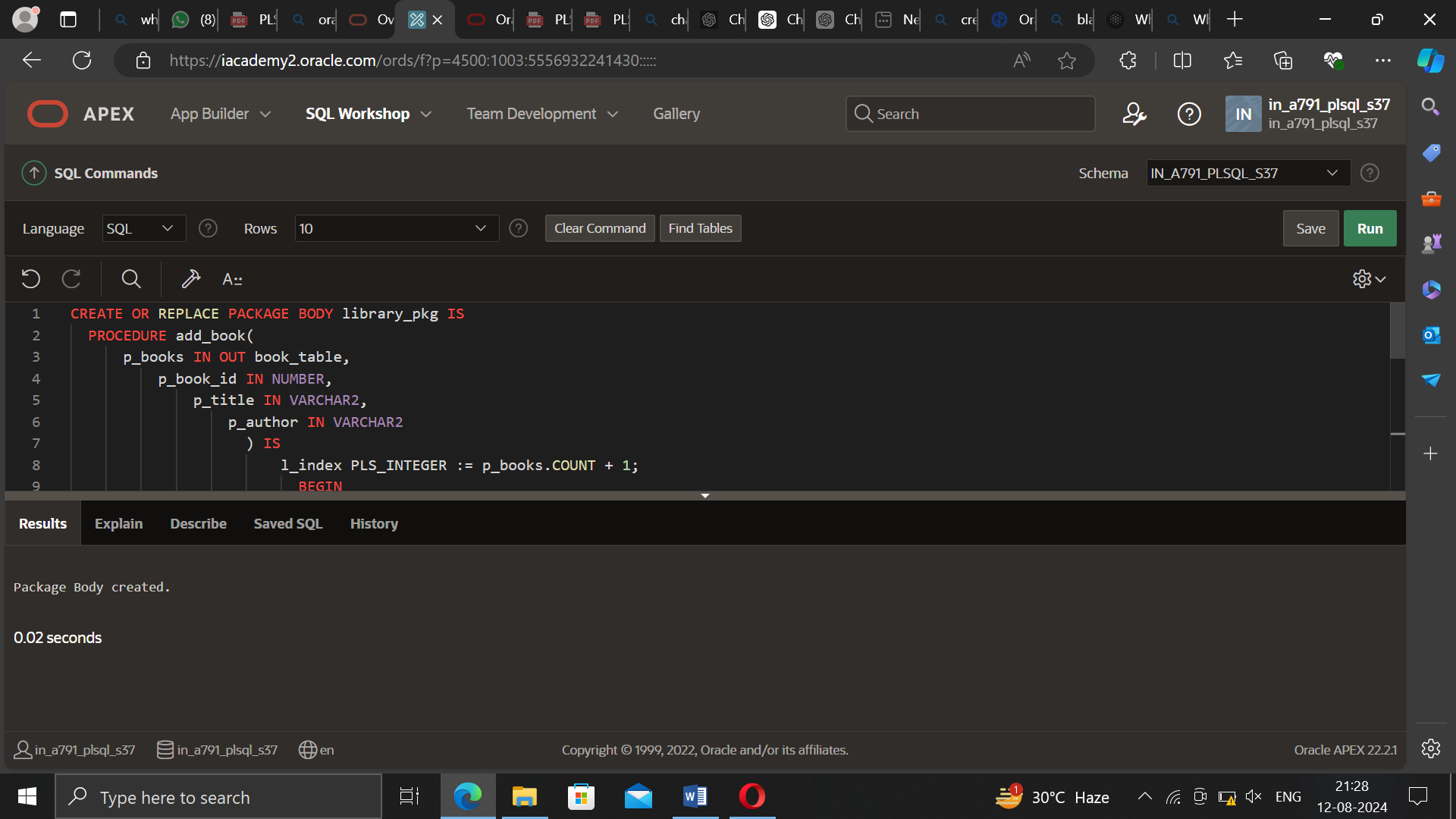
p\_author IN VARCHAR2

);

PROCEDURE list\_books(p\_books IN book\_table);

END library\_pkg;

/



CREATE OR REPLACE PACKAGE BODY library\_pkg IS

PROCEDURE add\_book(

p\_books IN OUT book\_table,

p\_book\_id IN NUMBER,

p\_title IN VARCHAR2,

p\_author IN VARCHAR2

) IS

l\_index PLS\_INTEGER := p\_books.COUNT + 1;

BEGIN

p\_books(l\_index) := book\_rec(p\_book\_id, p\_title, p\_author);

END add\_book;

PROCEDURE list\_books(p\_books IN book\_table) IS

BEGIN

FOR i IN 1..p\_books.COUNT LOOP

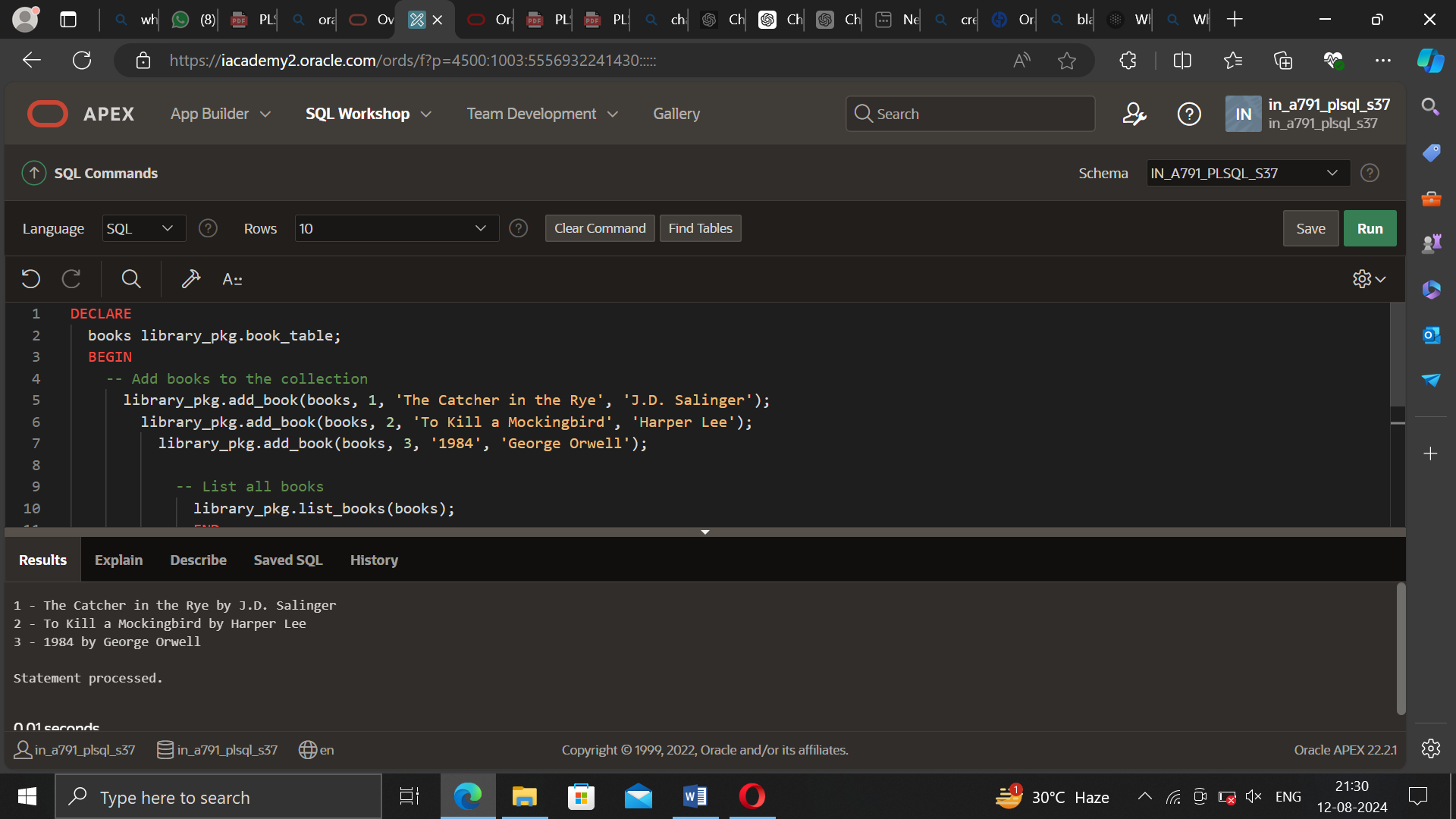
DBMS\_OUTPUT.PUT\_LINE(p\_books(i).book\_id || ' - ' || p\_books(i).title || ' by ' || p\_books(i).author);

END LOOP;

END list\_books;

END library\_pkg;

/



DECLARE

books library\_pkg.book\_table;

BEGIN

-- Add books to the collection

library\_pkg.add\_book(books, 1, 'The Catcher in the Rye', 'J.D. Salinger');

library\_pkg.add\_book(books, 2, 'To Kill a Mockingbird', 'Harper Lee');

library\_pkg.add\_book(books, 3, '1984', 'George Orwell');

-- List all books

library\_pkg.list\_books(books);

END;

/