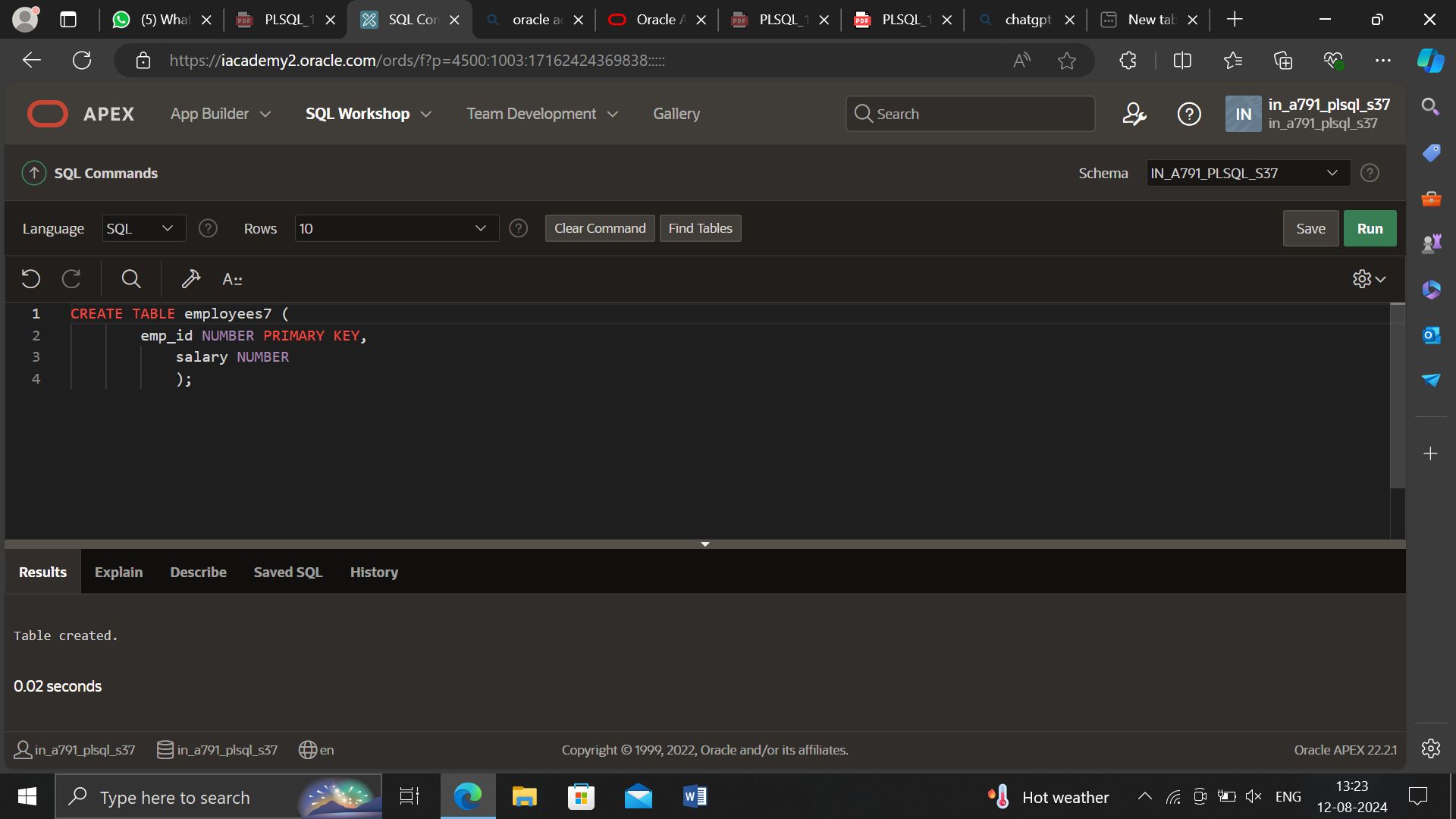
CREATE TABLE employees (

emp\_id NUMBER PRIMARY KEY,

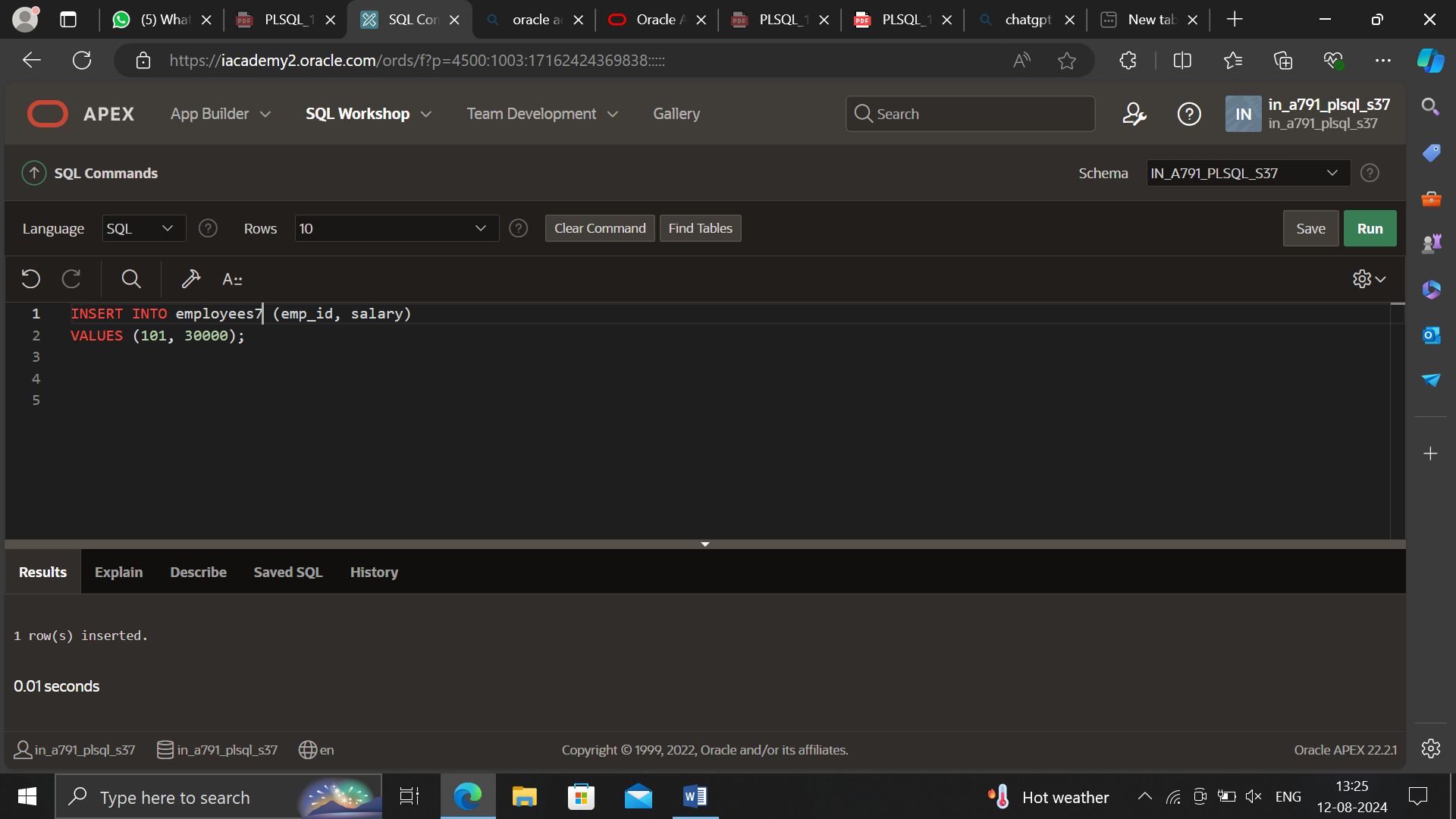
salary NUMBER

);



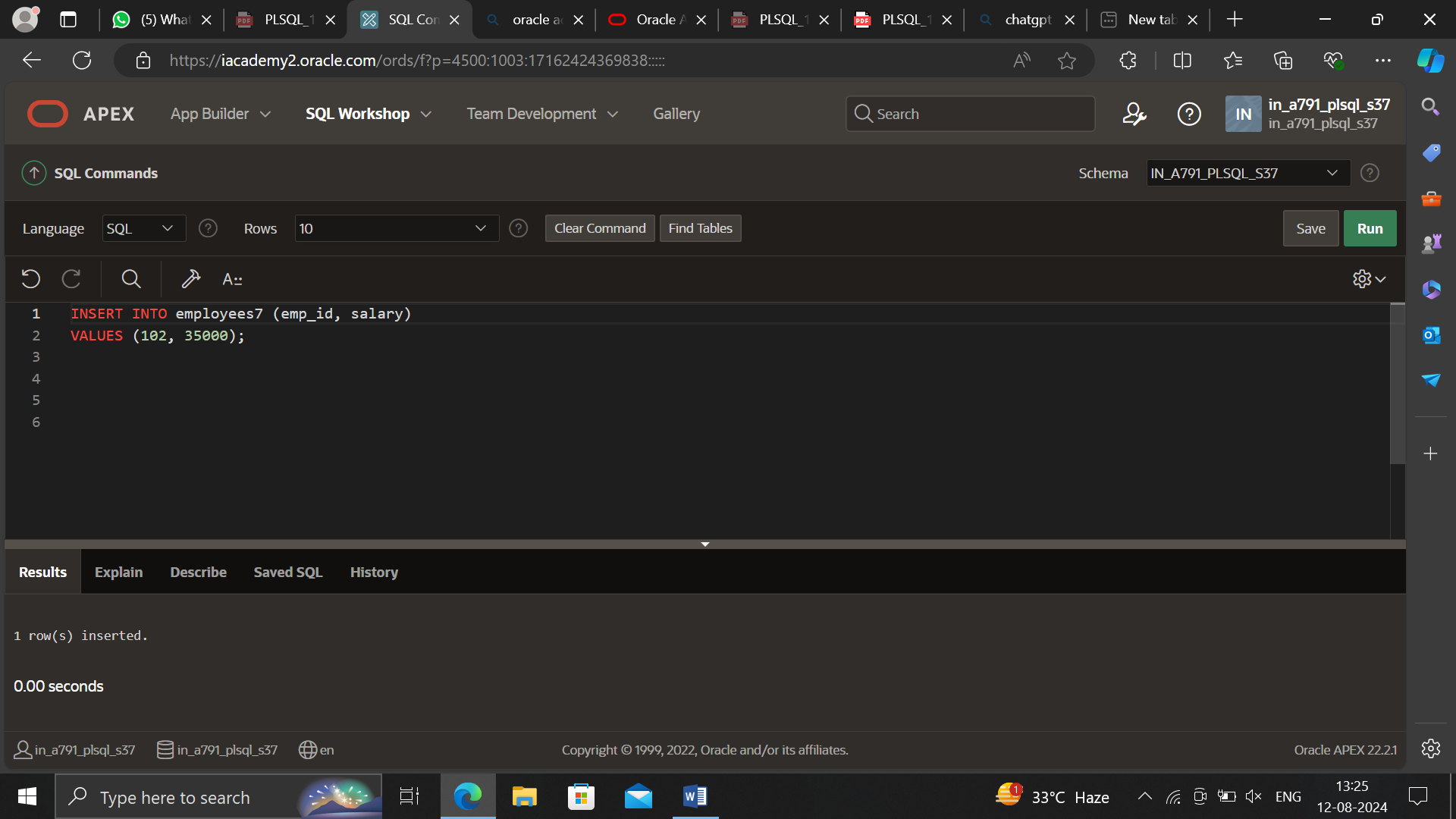
INSERT INTO employees (emp\_id, salary)

VALUES (101, 30000);



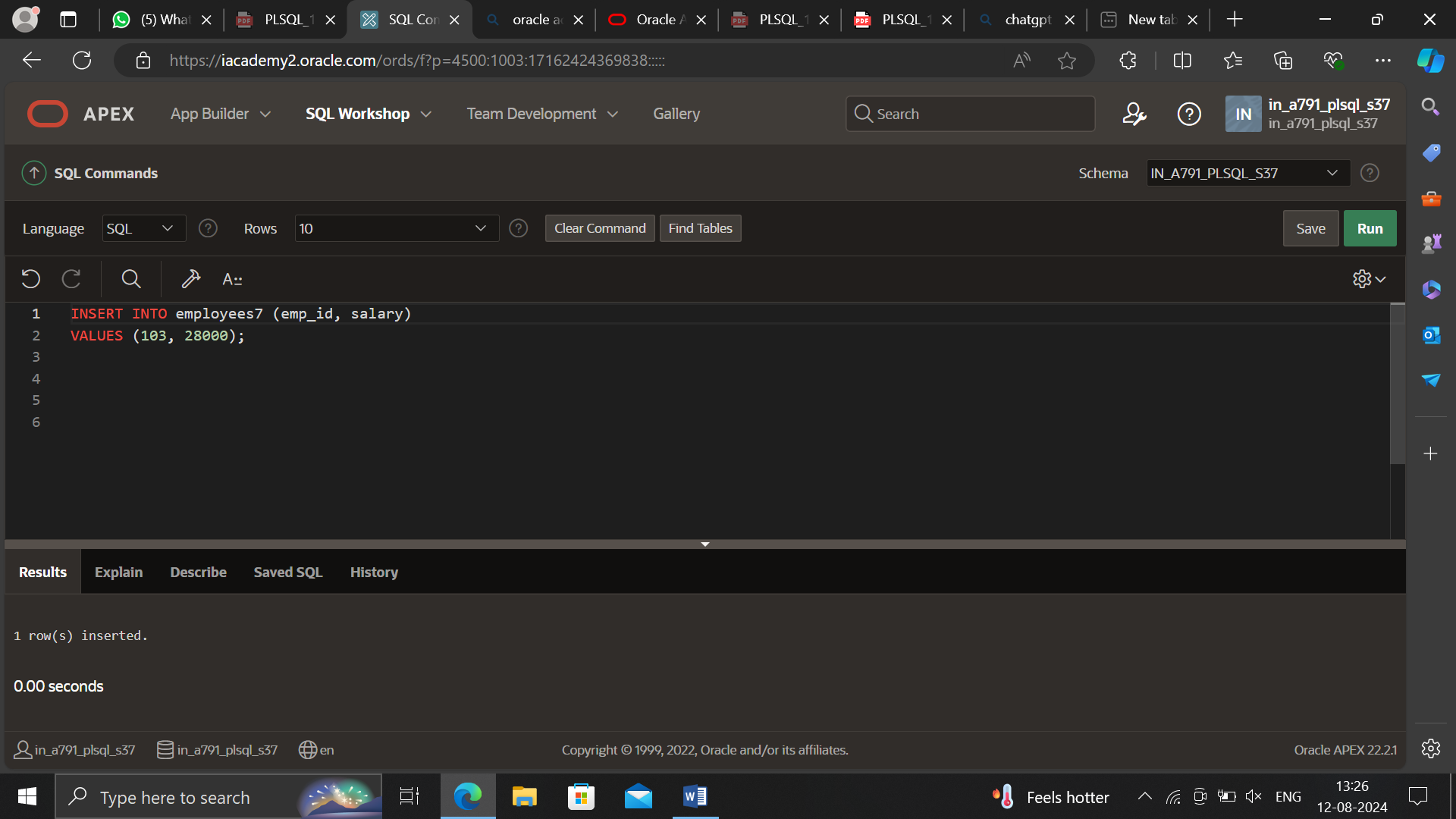
INSERT INTO employees (emp\_id, salary)

VALUES (102, 35000);



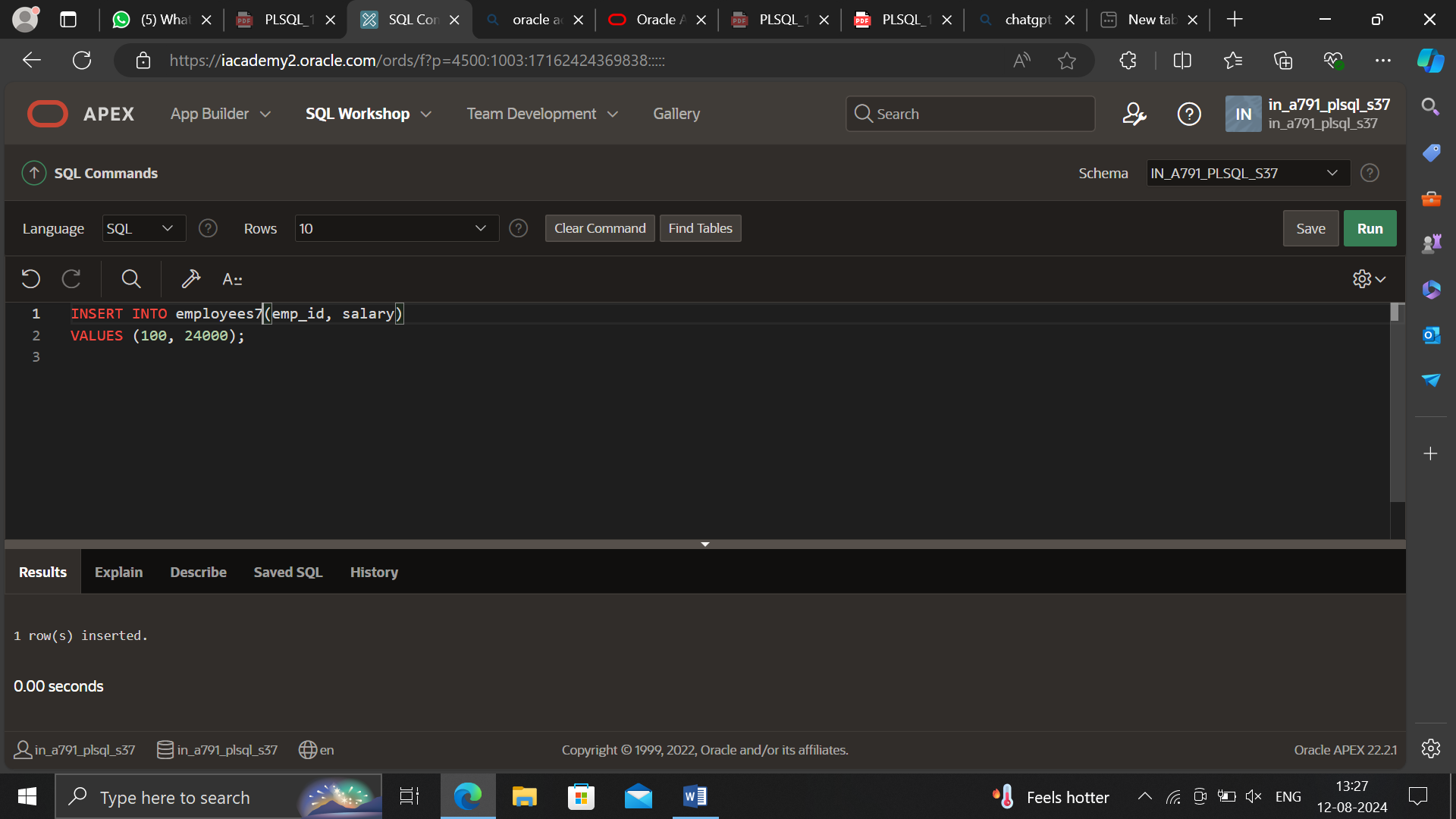
INSERT INTO employees (emp\_id, salary)

VALUES (103, 28000);



INSERT INTO employees (emp\_id, salary)

VALUES (100, 24000);



CREATE OR REPLACE PACKAGE salary\_pkg IS

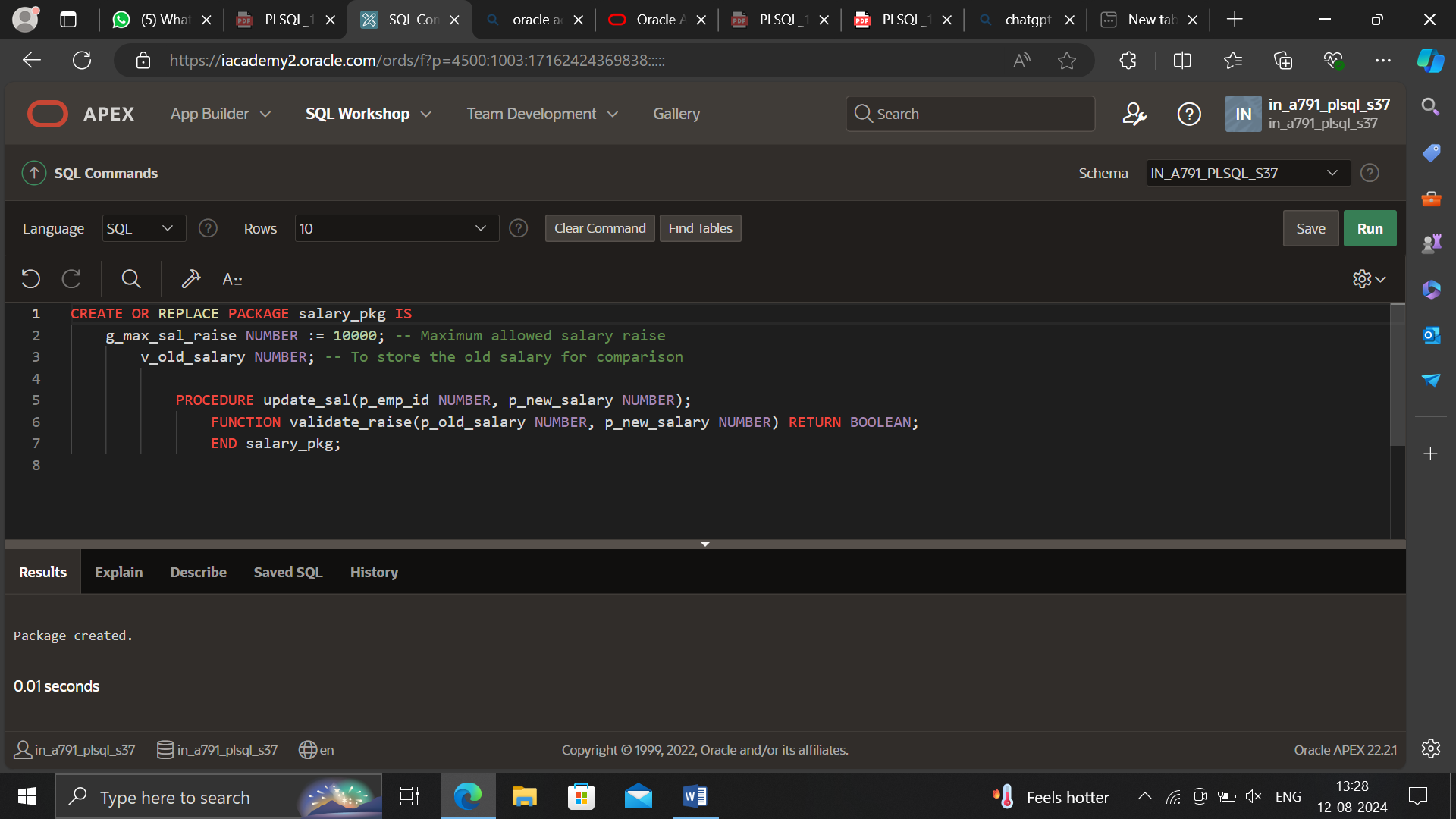
g\_max\_sal\_raise NUMBER := 10000; -- Maximum allowed salary raise

v\_old\_salary NUMBER; -- To store the old salary for comparison

PROCEDURE update\_sal(p\_emp\_id NUMBER, p\_new\_salary NUMBER);

FUNCTION validate\_raise(p\_old\_salary NUMBER, p\_new\_salary NUMBER) RETURN BOOLEAN;

END salary\_pkg;



CREATE OR REPLACE PACKAGE BODY salary\_pkg IS

-- Procedure to update the salary of an employee

PROCEDURE update\_sal(p\_emp\_id NUMBER, p\_new\_salary NUMBER) IS

BEGIN

SELECT salary INTO v\_old\_salary FROM employees WHERE emp\_id = p\_emp\_id;

UPDATE employees

SET salary = p\_new\_salary

WHERE emp\_id = p\_emp\_id;

END update\_sal;

-- Function to validate if the raise is within allowed limits

FUNCTION validate\_raise(p\_old\_salary NUMBER, p\_new\_salary NUMBER) RETURN BOOLEAN IS

BEGIN

IF (p\_new\_salary - p\_old\_salary) <= g\_max\_sal\_raise THEN

RETURN TRUE;

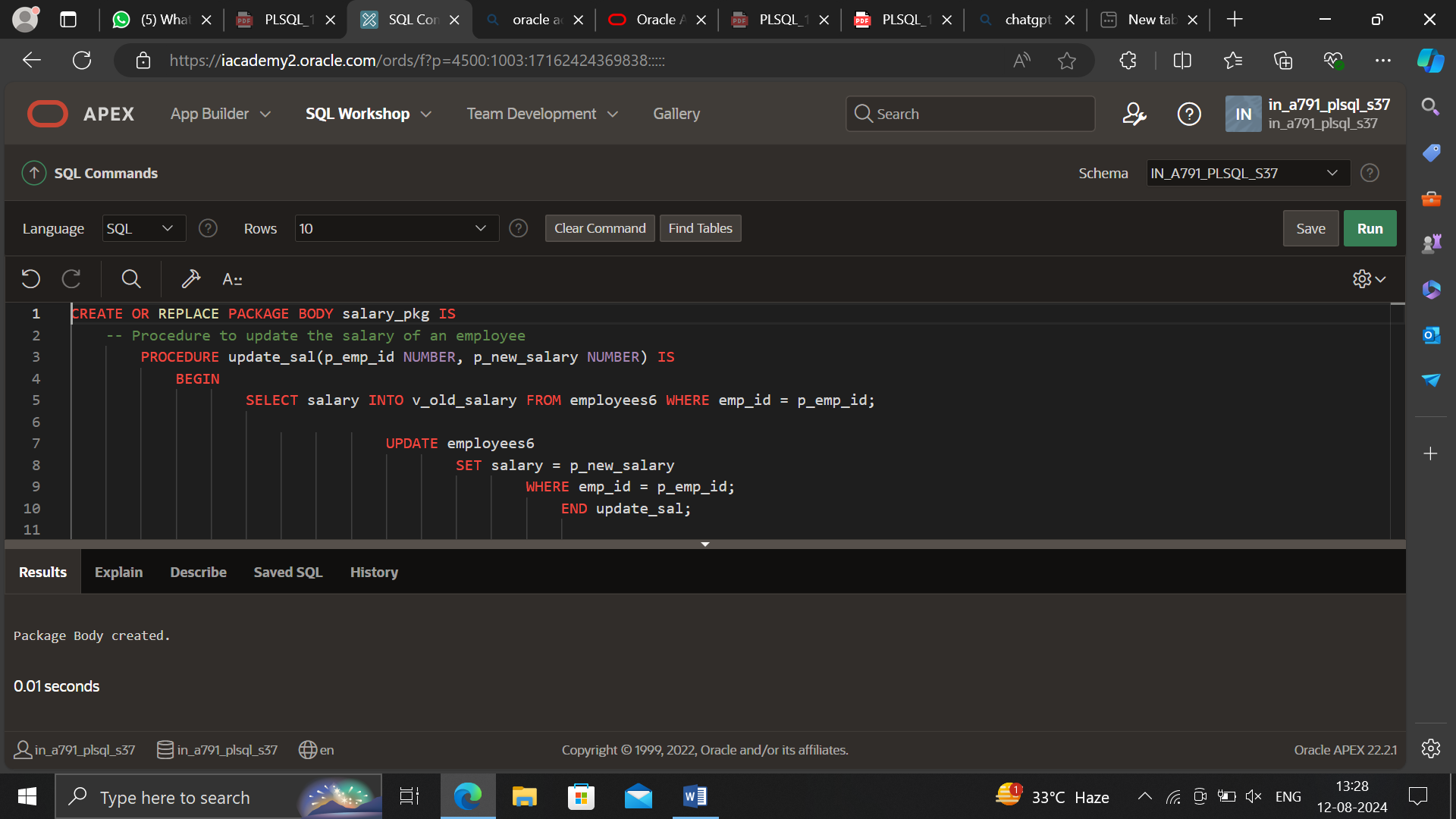
ELSE

RETURN FALSE;

END IF;

END validate\_raise;

END salary\_pkg;



DECLARE

v\_bool BOOLEAN;

v\_number NUMBER;

BEGIN

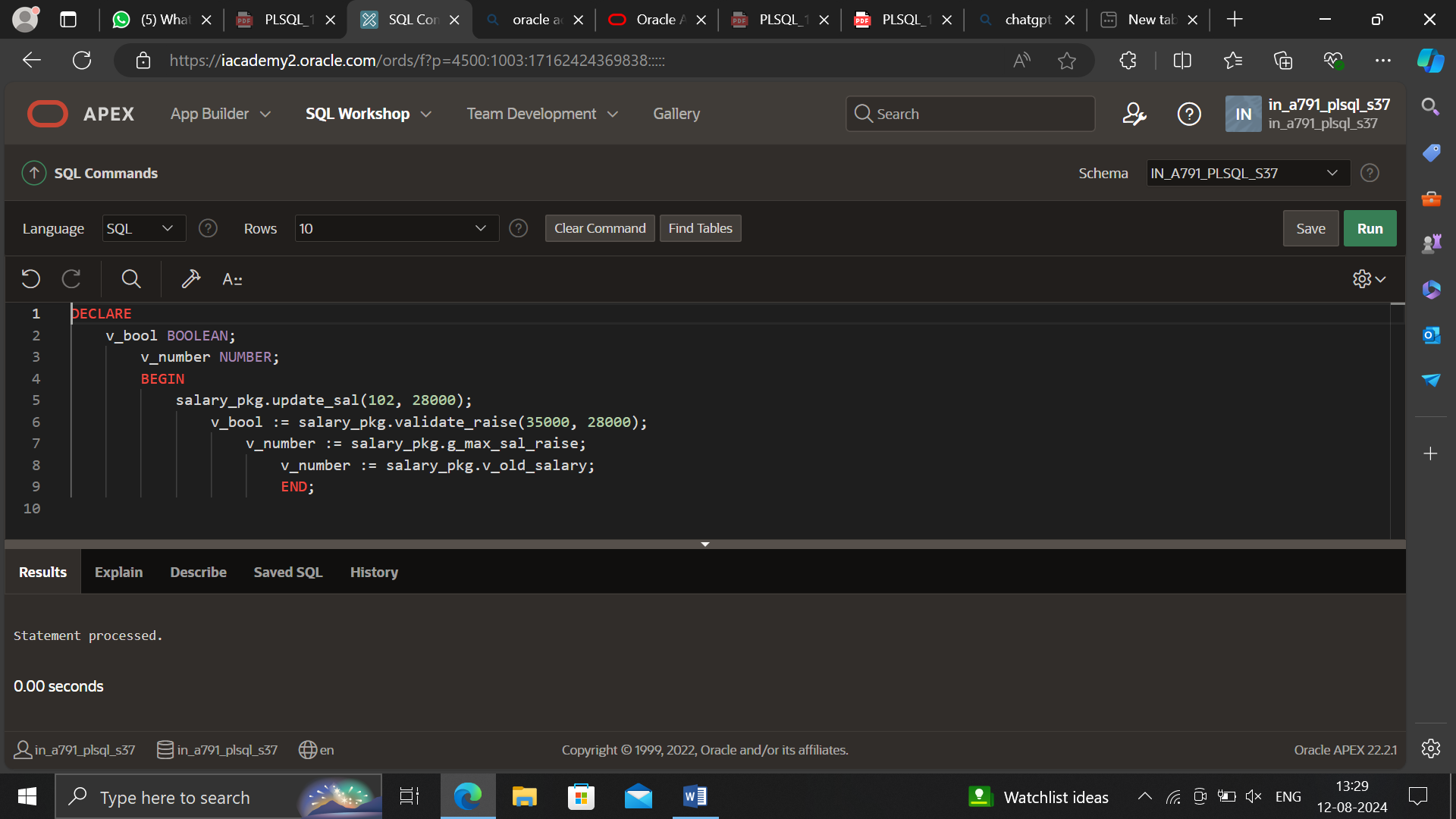
salary\_pkg.update\_sal(100, 25000); -- Updates the salary of employee 100 to 25000

v\_bool := salary\_pkg.validate\_raise(24000, 25000); -- Checks if the raise from 24000 to 25000 is valid

v\_number := salary\_pkg.g\_max\_sal\_raise; -- Retrieves the maximum salary raise allowed

v\_number := salary\_pkg.v\_old\_salary; -- Retrieves the old salary before the update

END;



Select \* from employee;

