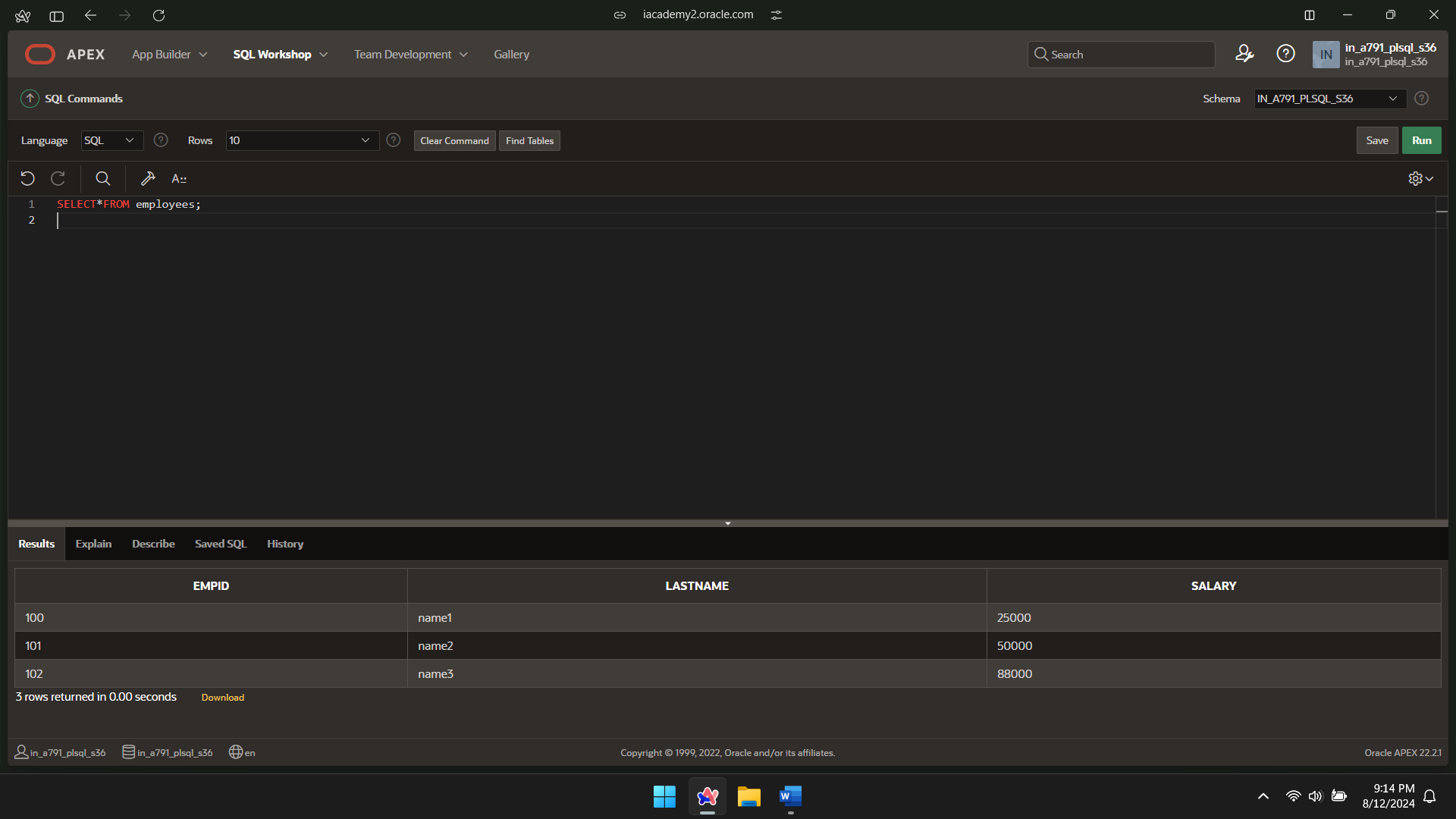
SELECT\*FROM employees;



PACKAGE

CREATE OR REPLACE PACKAGE salary\_pkg

IS

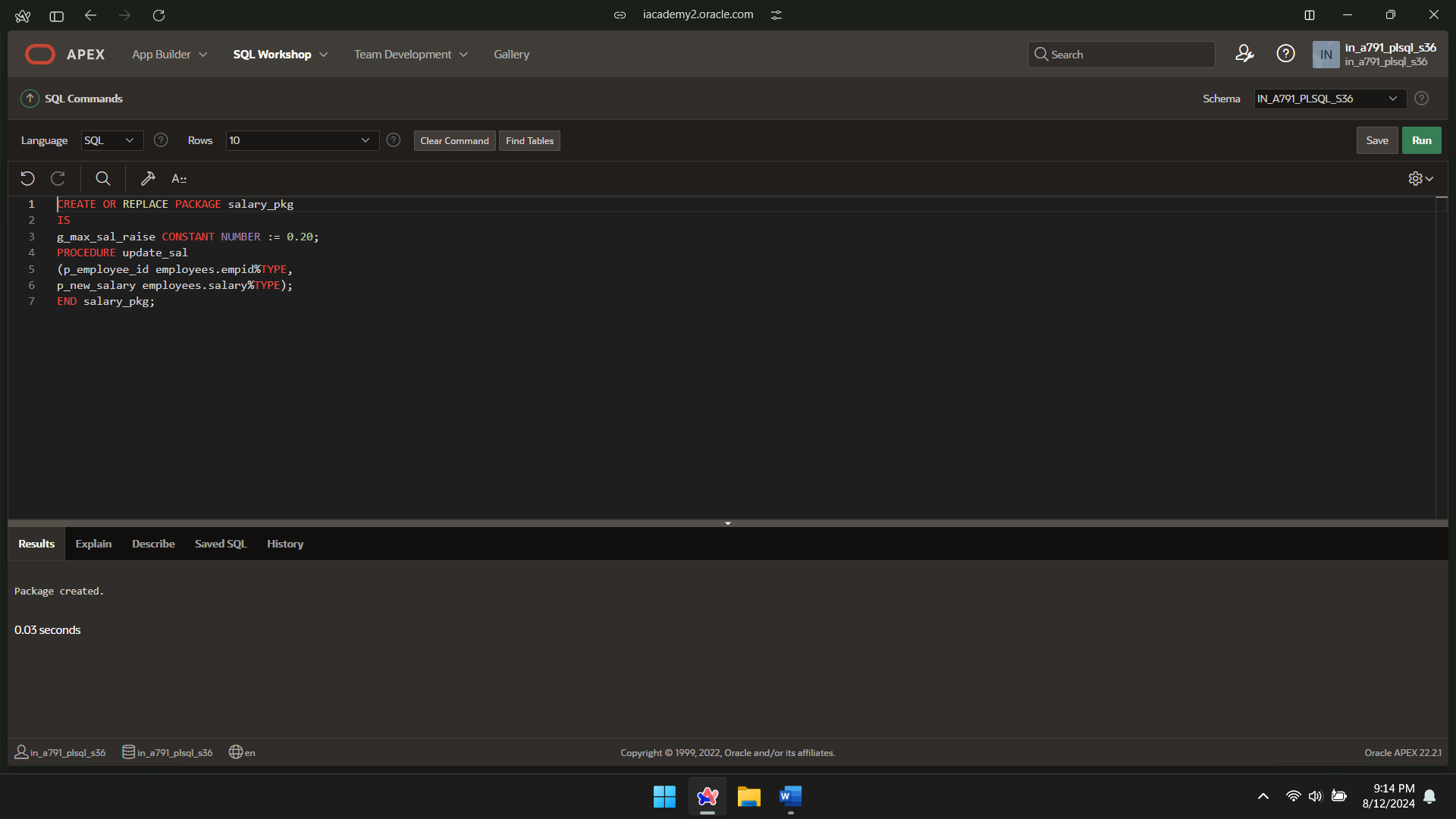
g\_max\_sal\_raise CONSTANT NUMBER := 0.20;

PROCEDURE update\_sal

(p\_employee\_id employees.empid%TYPE,

p\_new\_salary employees.salary%TYPE);

END salary\_pkg;



PACKAGE BODY

CREATE OR REPLACE PACKAGE BODY salary\_pkg IS

FUNCTION validate\_raise

(p\_old\_salary employees.salary%TYPE,

p\_new\_salary employees.salary%TYPE)

RETURN BOOLEAN IS

BEGIN

IF p\_new\_salary >

(p\_old\_salary \* (1 + g\_max\_sal\_raise)) THEN

RETURN FALSE;

ELSE

RETURN TRUE;

END IF;

END validate\_raise;

PROCEDURE update\_sal

(p\_employee\_id employees.empid%TYPE,

p\_new\_salary employees.salary%TYPE)

IS v\_old\_salary employees.salary%TYPE;

BEGIN

SELECT salary INTO v\_old\_salary FROM employees

WHERE empid = p\_employee\_id;

IF validate\_raise(v\_old\_salary, p\_new\_salary) THEN

UPDATE employees

SET salary = p\_new\_salary

WHERE empid = p\_employee\_id;

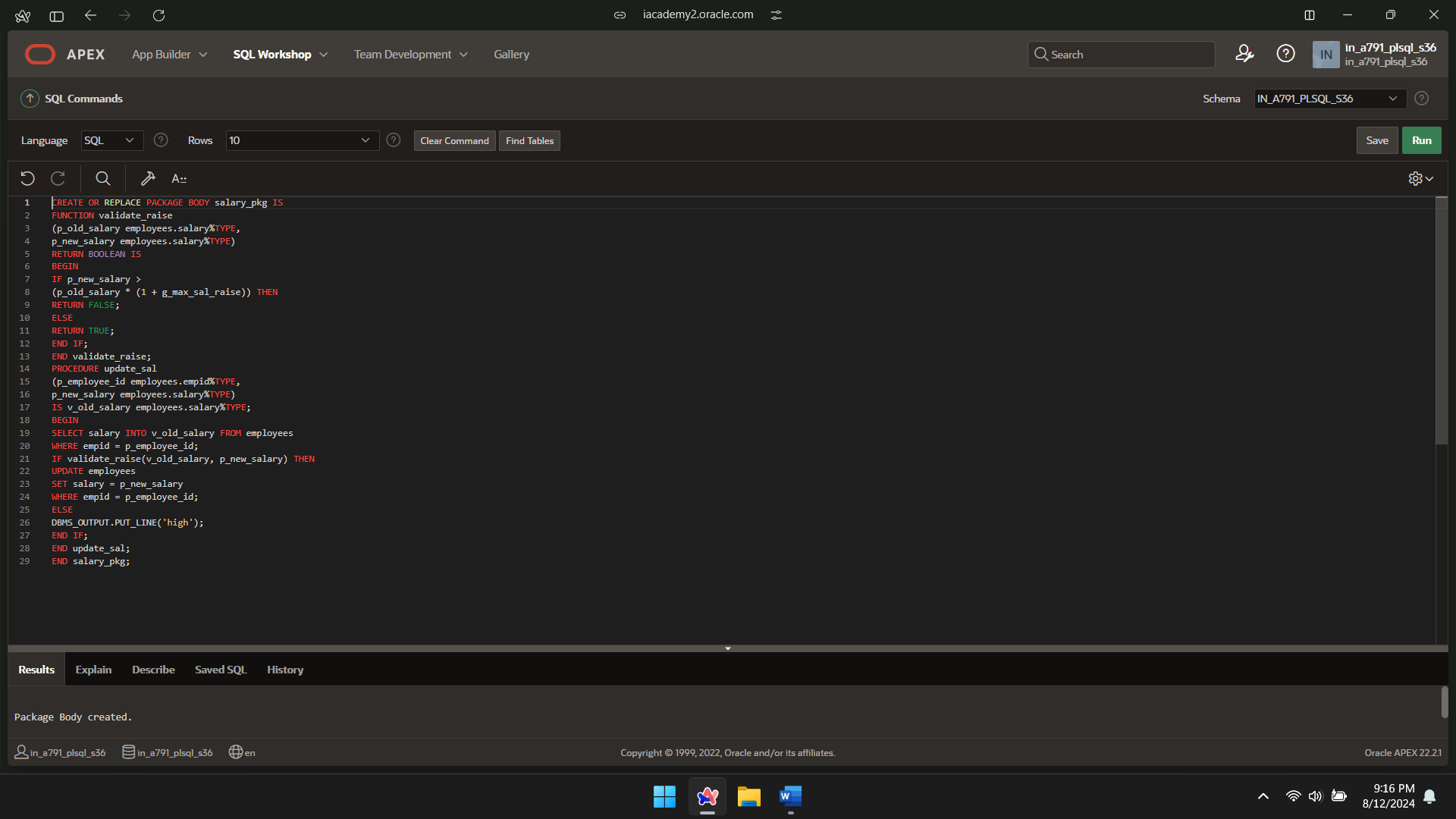
ELSE

DBMS\_OUTPUT.PUT\_LINE('high');

END IF;

END update\_sal;

END salary\_pkg;



INVOKE PACKAGE

DECLARE

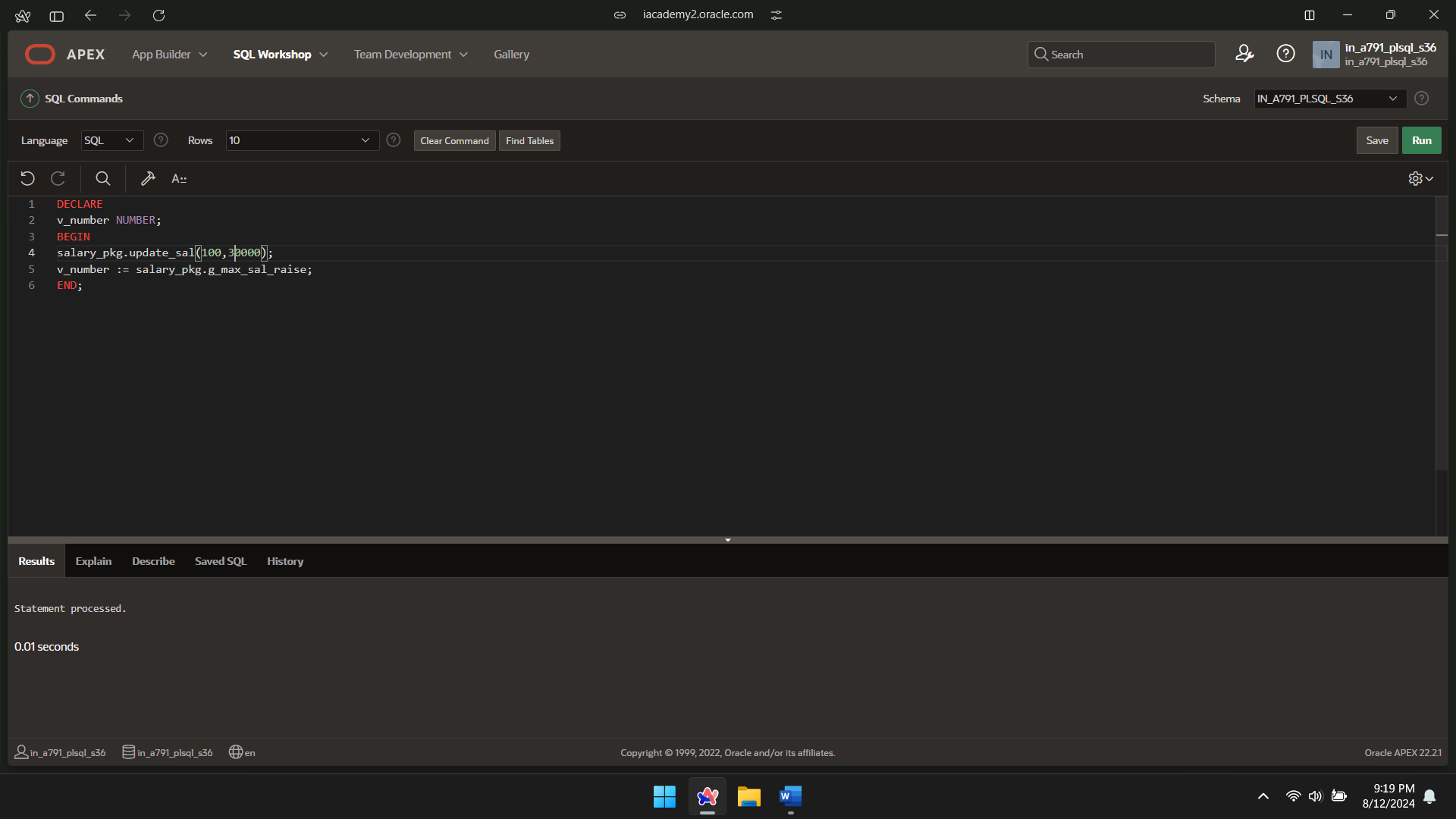
v\_number NUMBER;

BEGIN

salary\_pkg.update\_sal(100,30000);

v\_number := salary\_pkg.g\_max\_sal\_raise;

END;



SELECT\*FROM employees;

