**Class Test 03**

**PL/SQL**

**Part 01:**

1. Write a query that can multiply two numbers taking input from user.

Query: DECLARE

num1 NUMBER;

num2 NUMBER;

result NUMBER;

BEGIN

num1 := 15;

num2 :=30;

result := num1 \* num2;

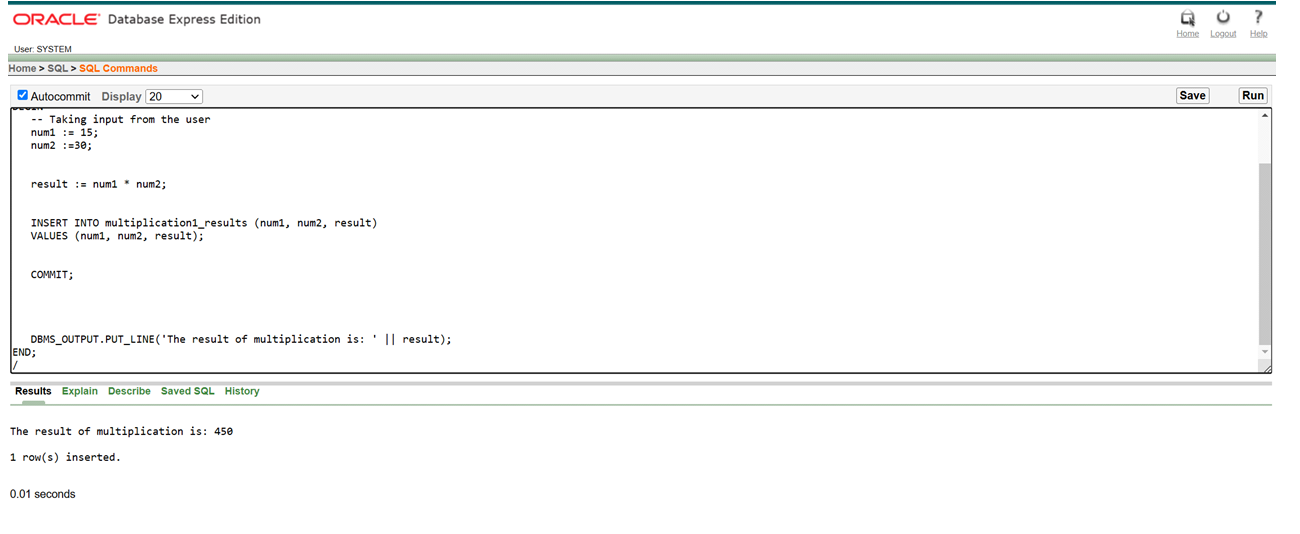
INSERT INTO multiplication1\_results (num1, num2, result)

VALUES (num1, num2, result);

COMMIT;

DBMS\_OUTPUT.PUT\_LINE('The result of multiplication is: ' || result);

END;



1. Write a query that can add two numbers if the numbers are equal. Use CASE Statement.

Query: DECLARE

a NUMBER := 9;

b NUMBER := 10;

result NUMBER;

BEGIN

result := CASE

WHEN a = b THEN a + b

ELSE NULL

END;

IF result IS NOT NULL THEN

DBMS\_OUTPUT.PUT\_LINE('Sum: ' || result);

ELSE

DBMS\_OUTPUT.PUT\_LINE('Numbers are not equal. No addition performed.');

END IF;

END;

A screenshot of a computer

AI-generated content may be incorrect.

1. Write a query that can check if two strings are equal or not. Use IF-THEN-ELSIF Statement.

DECLARE

str1 VARCHAR2(20) := 'Hello';

str2 VARCHAR2(20) := 'Hello';

BEGIN

IF str1 = str2 THEN

DBMS\_OUTPUT.PUT\_LINE('Strings are equal.');

ELSIF str1 != str2 THEN

DBMS\_OUTPUT.PUT\_LINE('Strings are NOT equal.');

END IF;

END;

A screenshot of a computer

AI-generated content may be incorrect.

1. Write a query that can multiply two numbers. If the result obtained is less than 100, **Hi** is displayed, if the result obtained is more than 100, **Bye** is displayed and if the result obtained is equal to 100, **ADBMS** is displayed. Use IF-THEN-ELSIF Statement

Query: DECLARE

a NUMBER := 9;

b NUMBER := 10;

result NUMBER;

BEGIN

result := a \* b;

IF result < 100 THEN

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

ELSIF result > 100 THEN

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

ELSIF result = 100 THEN

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

END IF;

END;

A screenshot of a computer

AI-generated content may be incorrect.

1. Write a query that can check if two numbers are equal or not. Use CASE Statement.

Query: DECLARE

x NUMBER := 60;

y NUMBER := 50;

msg VARCHAR2(50);

BEGIN

msg := CASE

WHEN x = y THEN 'Numbers are Equal'

ELSE 'Numbers are NOT Equal'

END;

DBMS\_OUTPUT.PUT\_LINE(msg);

END;

A screenshot of a computer

AI-generated content may be incorrect.

**Part 02:**

*To solve the following use the scott schema*

1. Write a query that can display the salary of employee ALLEN. If ALLEN’s salary is greater than 2000 display ‘SALARY GREATER THAN 2000’ and If not then display ‘SALARY LESS THAN 2000’.

Query: SELECT

ename,

sal,

CASE

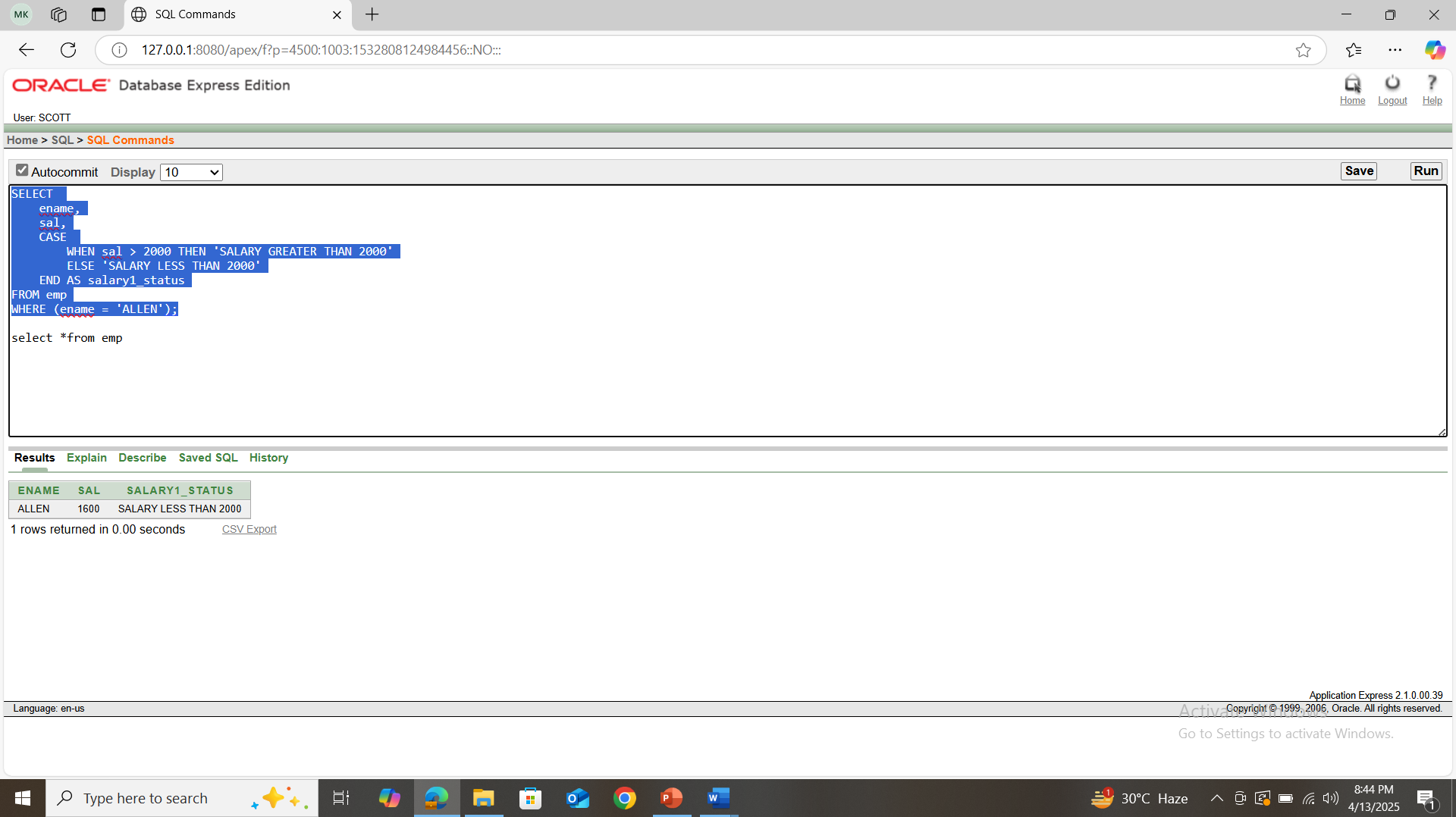
WHEN sal > 2000 THEN 'SALARY GREATER THAN 2000'

ELSE 'SALARY LESS THAN 2000'

END AS salary1\_status

FROM emp

WHERE (ename = 'ALLEN');



1. Write a query that can ask user to input the EMPNO of employee WARD and display his salary.

Query: SELECT ename, sal

FROM emp

WHERE (empno = 7521);

A screenshot of a computer

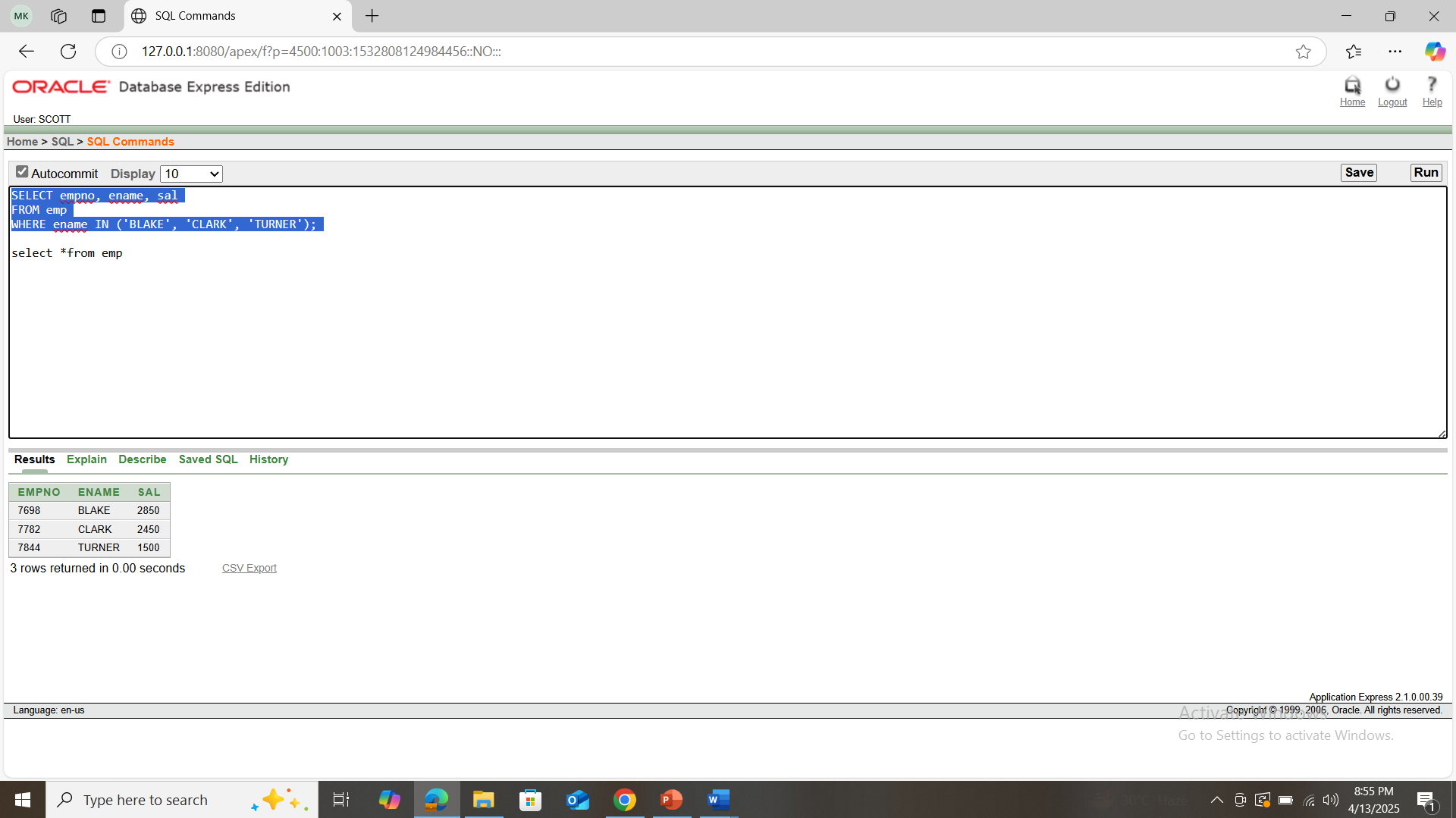
AI-generated content may be incorrect.

1. Write a query that can ask user to input the EMPNO of employee BLAKE,CLARK and TURNER and display their respective salary.

Query: SELECT empno, ename, sal

FROM emp

WHERE ename IN ('BLAKE', 'CLARK', 'TURNER');

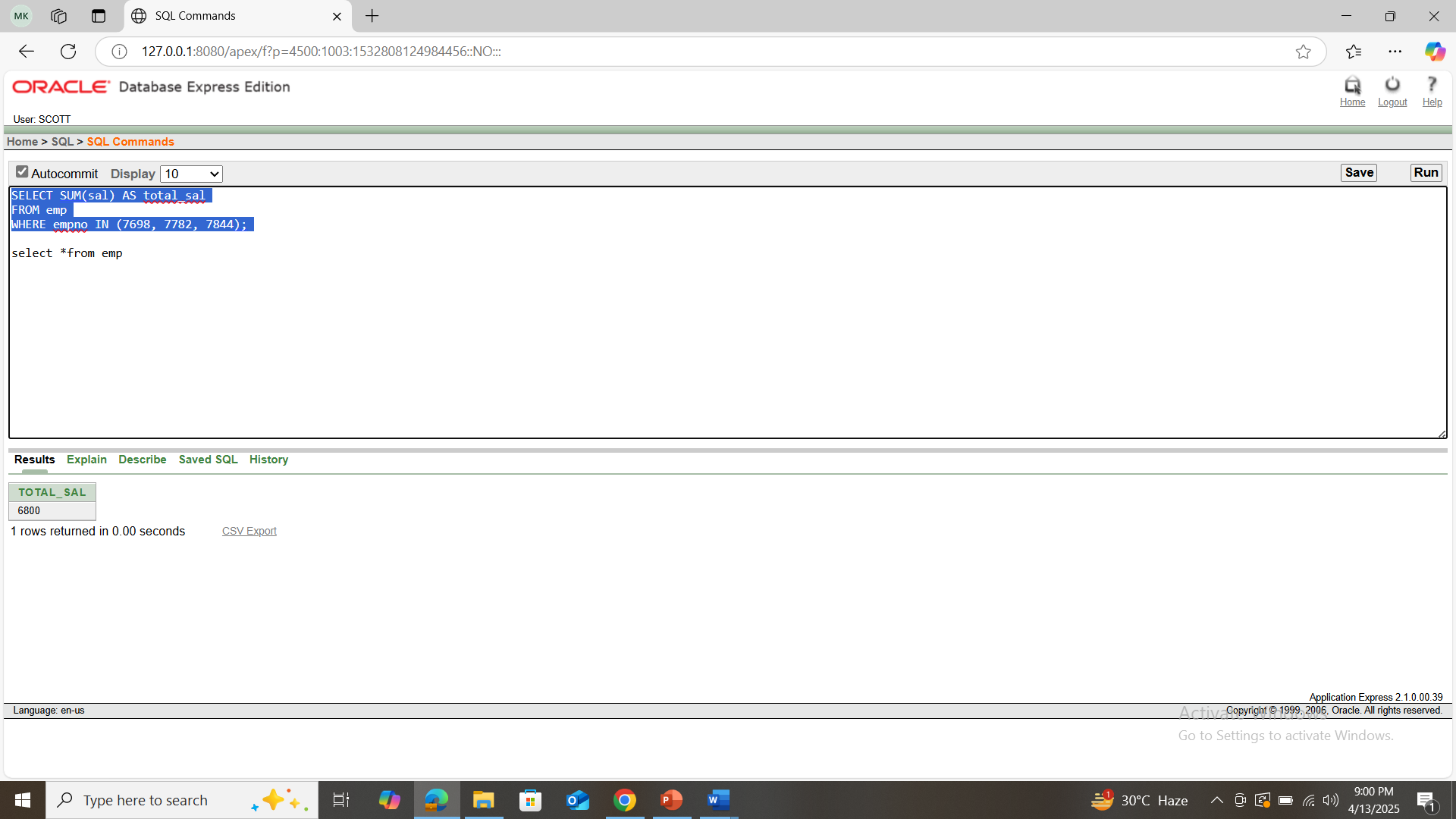


1. Write a query that can ask user to input the EMPNO of employee BLAKE,CLARK and TURNER and display their respective salary, add the salaries and display the total.

Query: SELECT SUM(sal) AS total\_sal

FROM emp

WHERE empno IN (7698, 7782, 7844);

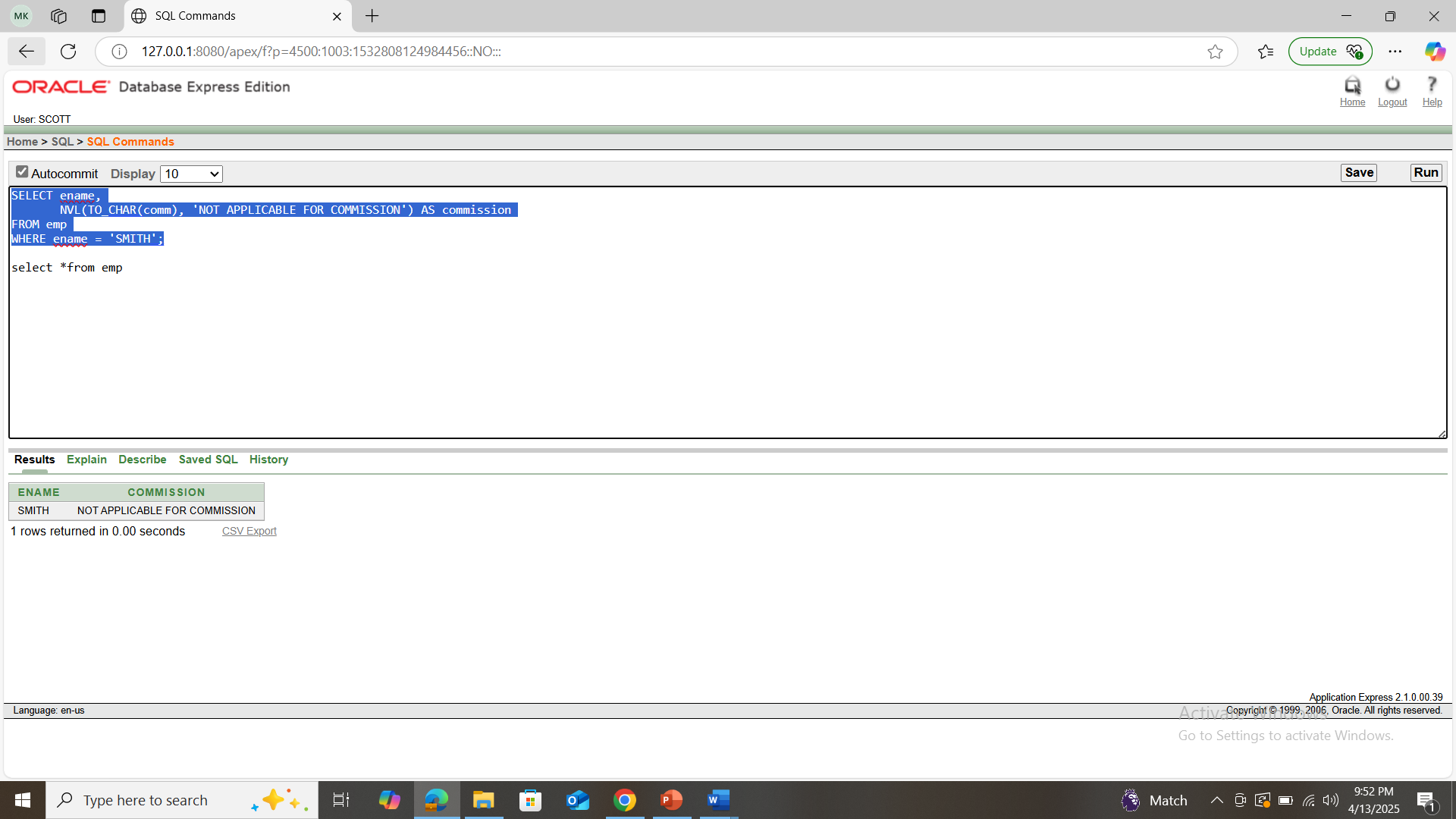
1. 
2. Write a query that displays the commission of employee SMITH. If SMITH’s commission is NULL. Display ‘NOT APPLICABLE FOR COMMISSION’

Query: SELECT ename,

NVL(TO\_CHAR(comm), 'NOT APPLICABLE FOR COMMISSION') AS commission

FROM emp

WHERE ename = 'SMITH';



**Part 03:**

*To solve the following use the scott schema*

1. Write a query that can display the salary of employee JONES three times using basic loop.

DECLARE

v\_sal emp.sal%TYPE;

i NUMBER := 1;

BEGIN

SELECT sal INTO v\_sal FROM emp WHERE ename = 'JONES';

LOOP

EXIT WHEN i > 3;

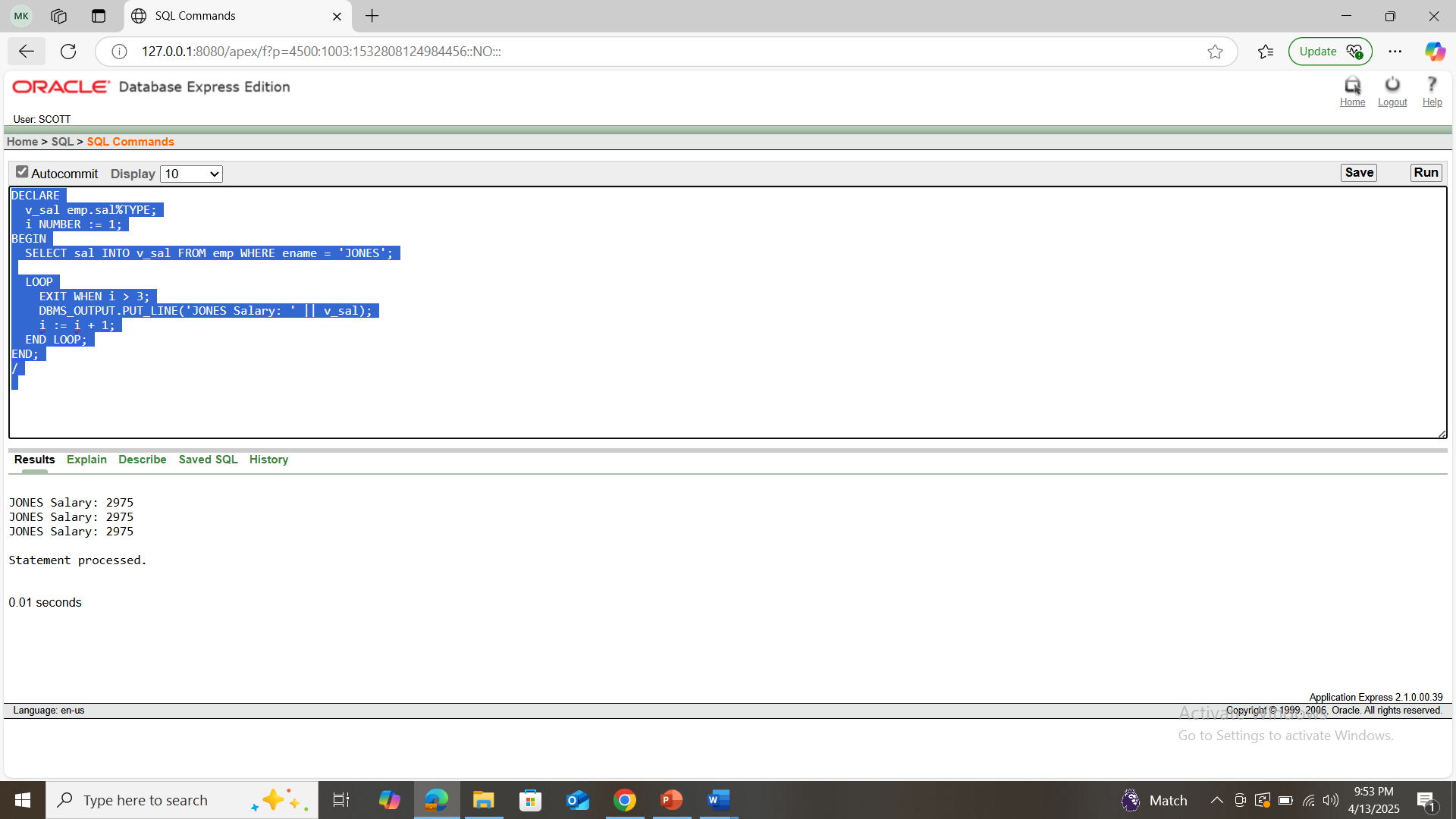
DBMS\_OUTPUT.PUT\_LINE('JONES Salary: ' || v\_sal);

i := i + 1;

END LOOP;

END;

/



1. Write a query that can display the salary of employee JONES three times using while loop.

Query: DECLARE

v\_sal emp.sal%TYPE;

i NUMBER := 1;

BEGIN

SELECT sal INTO v\_sal FROM emp WHERE ename = 'JONES';

WHILE i <= 3 LOOP

DBMS\_OUTPUT.PUT\_LINE('JONES Salary: ' || v\_sal);

i := i + 1;

END LOOP;

END;

/

A screenshot of a computer

AI-generated content may be incorrect.

1. Write a query that can display the salary of employee JONES three times using for loop.

Query: DECLARE

v\_sal emp.sal%TYPE;

BEGIN

SELECT sal INTO v\_sal FROM emp WHERE ename = 'JONES';

FOR i IN 1..3 LOOP

DBMS\_OUTPUT.PUT\_LINE('JONES Salary: ' || v\_sal);

END LOOP;

END;

A screenshot of a computer

AI-generated content may be incorrect.

1. Create a function that returns the total number of departments.

Query: CREATE OR REPLACE FUNCTION total\_departments

RETURN NUMBER

IS

v\_count NUMBER;

BEGIN

SELECT COUNT(\*) INTO v\_count FROM dept;

RETURN v\_count;

END;

/

BEGIN

DBMS\_OUTPUT.PUT\_LINE('Total Departments: ' || total\_departments);

END;

A screenshot of a computer

AI-generated content may be incorrect.

1. Create a procedure to update the salary of employee Allen to 100.

Query: CREATE OR REPLACE PROCEDURE update\_allen\_salary

IS

BEGIN

UPDATE emp

SET sal = 100

WHERE ename = 'ALLEN';

COMMIT;

END;

/

BEGIN

update\_allen\_salary;

END;

select \*from emp

A screenshot of a computer

AI-generated content may be incorrect.

**\*\*After solving the above questions using Oracle 10g, write the PL/SQLs in a MS Word document (Write down the answer and give screenshot of the result of the query. The name of the document MUST be your ID and the PL/SQLs MUST be numbered accordingly) and upload it in the provided link in your VUES account**