**Class Test 03**

**PL/SQL**

**Part 01:**

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

**Answer:**

DECLARE

num1 NUMBER := :Enter\_Number1;

num2 NUMBER := :Enter\_Number2;

result NUMBER;

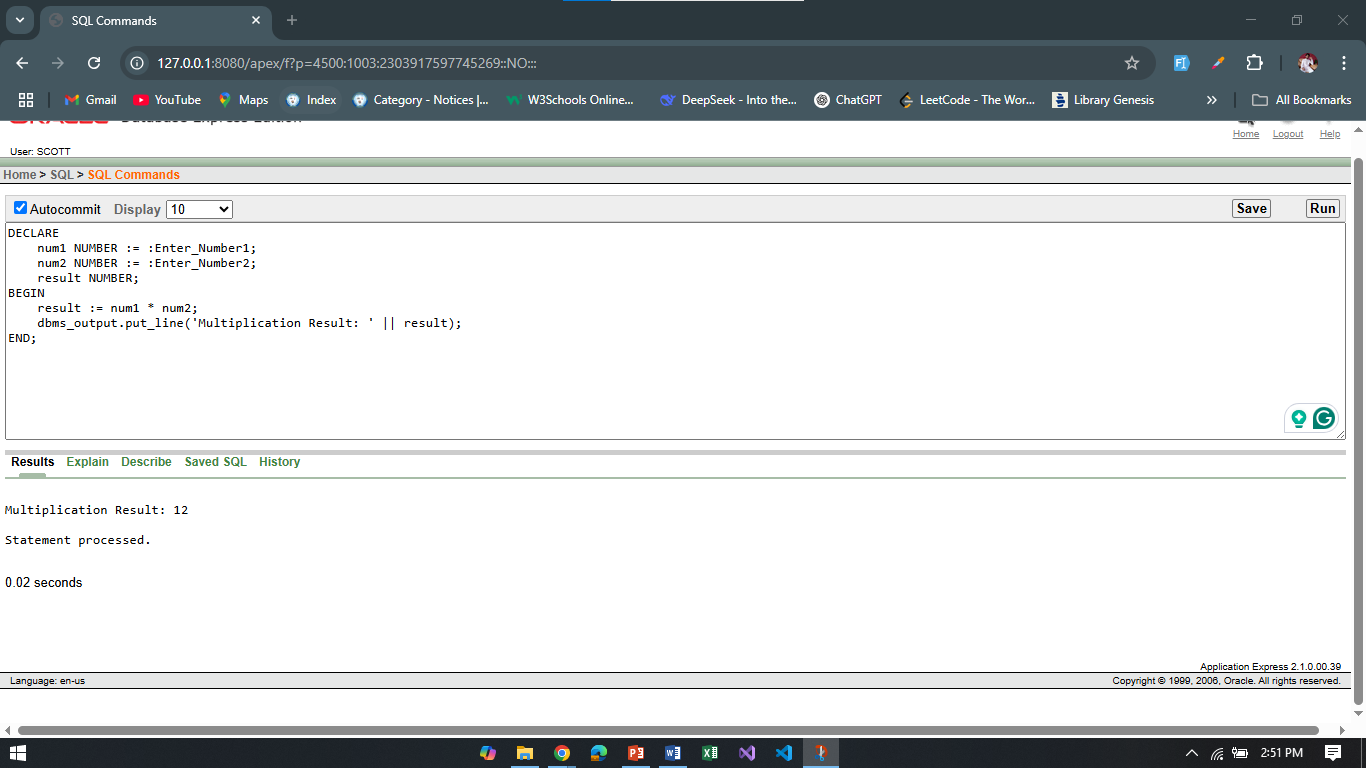
BEGIN

result := num1 \* num2;

dbms\_output.put\_line('Multiplication Result: ' || result);

END;

**Output:**



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

**Answer:**

DECLARE

num1 NUMBER := :Enter\_Number1;

num2 NUMBER := :Enter\_Number2;

result NUMBER;

BEGIN

result := CASE

WHEN num1 = num2 THEN num1 + num2

ELSE NULL

END;

IF result IS NOT NULL THEN

dbms\_output.put\_line('Sum of Numbers: ' || result);

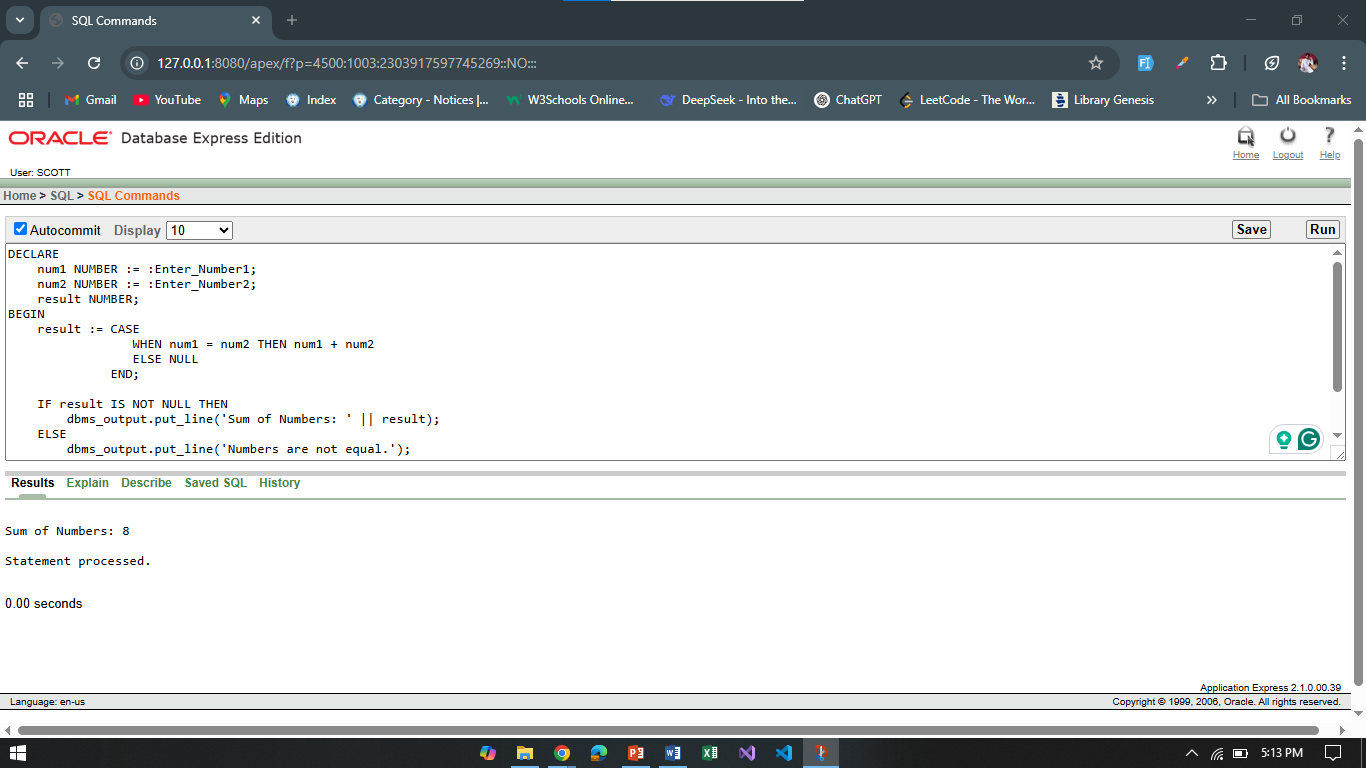
ELSE

dbms\_output.put\_line('Numbers are not equal.');

END IF;

END;

**Output:**



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

**Answer:**

DECLARE

str1 VARCHAR2(50) := 'String1';

str2 VARCHAR2(50) := 'String2';

BEGIN

IF str1 = str2 THEN

dbms\_output.put\_line('Strings are equal.');

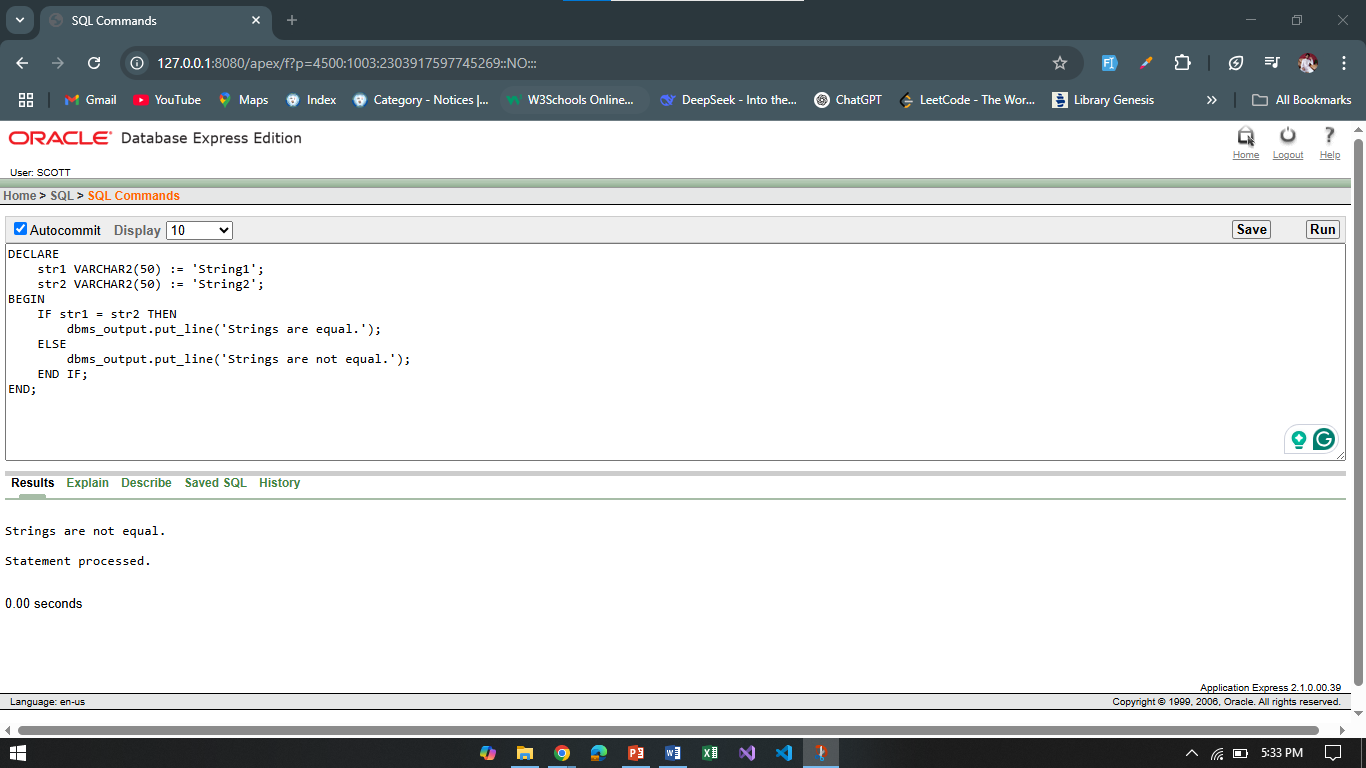
ELSE

dbms\_output.put\_line('Strings are not equal.');

END IF;

END;

**Output:**



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

**Answer:**

DECLARE

num1 NUMBER := :Enter\_Number1;

num2 NUMBER := :Enter\_Number2;

result NUMBER;

BEGIN

result := num1 \* num2;

IF result < 100 THEN

dbms\_output.put\_line('Hi');

ELSIF result > 100 THEN

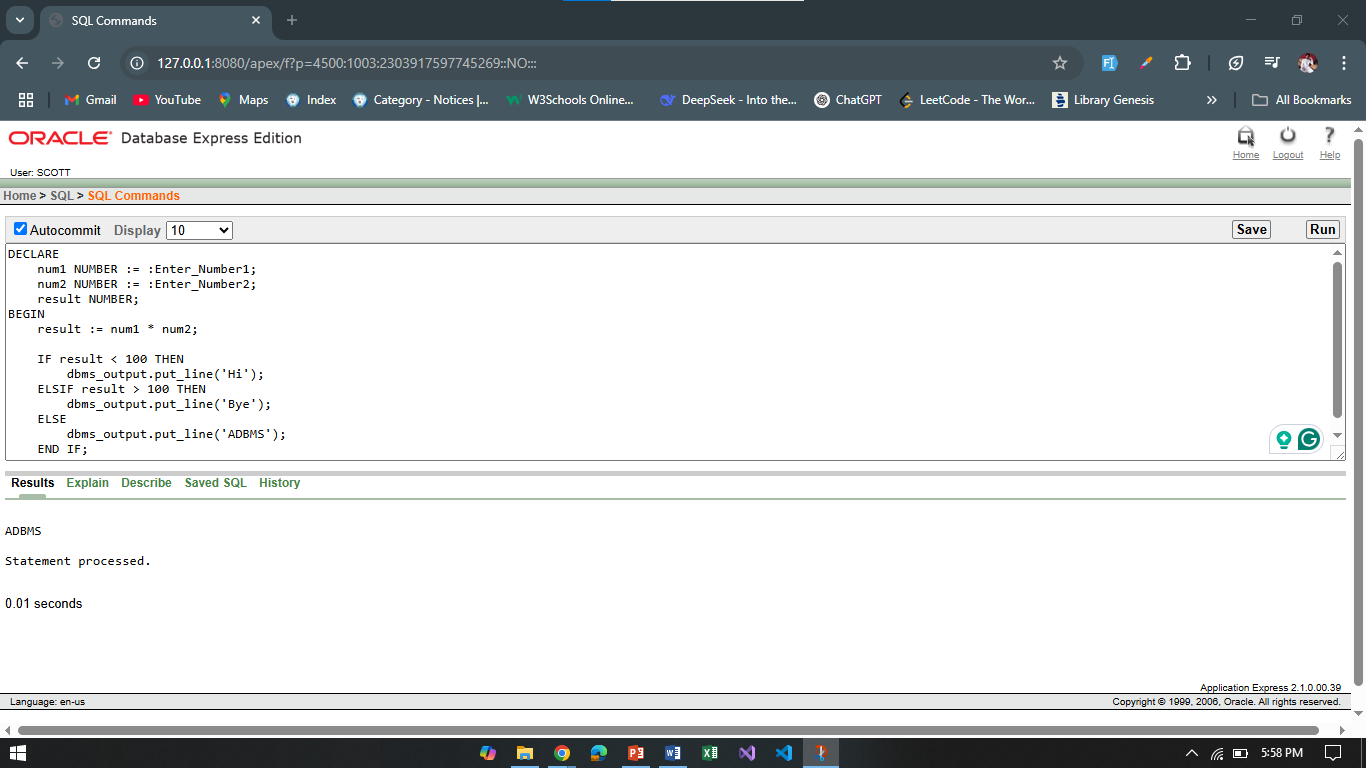
dbms\_output.put\_line('Bye');

ELSE

dbms\_output.put\_line('ADBMS');

END IF;

END;



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

**Answer:**

DECLARE

num1 NUMBER := :Enter\_Number1;

num2 NUMBER := :Enter\_Number2;

message VARCHAR2(50);

BEGIN

message := CASE

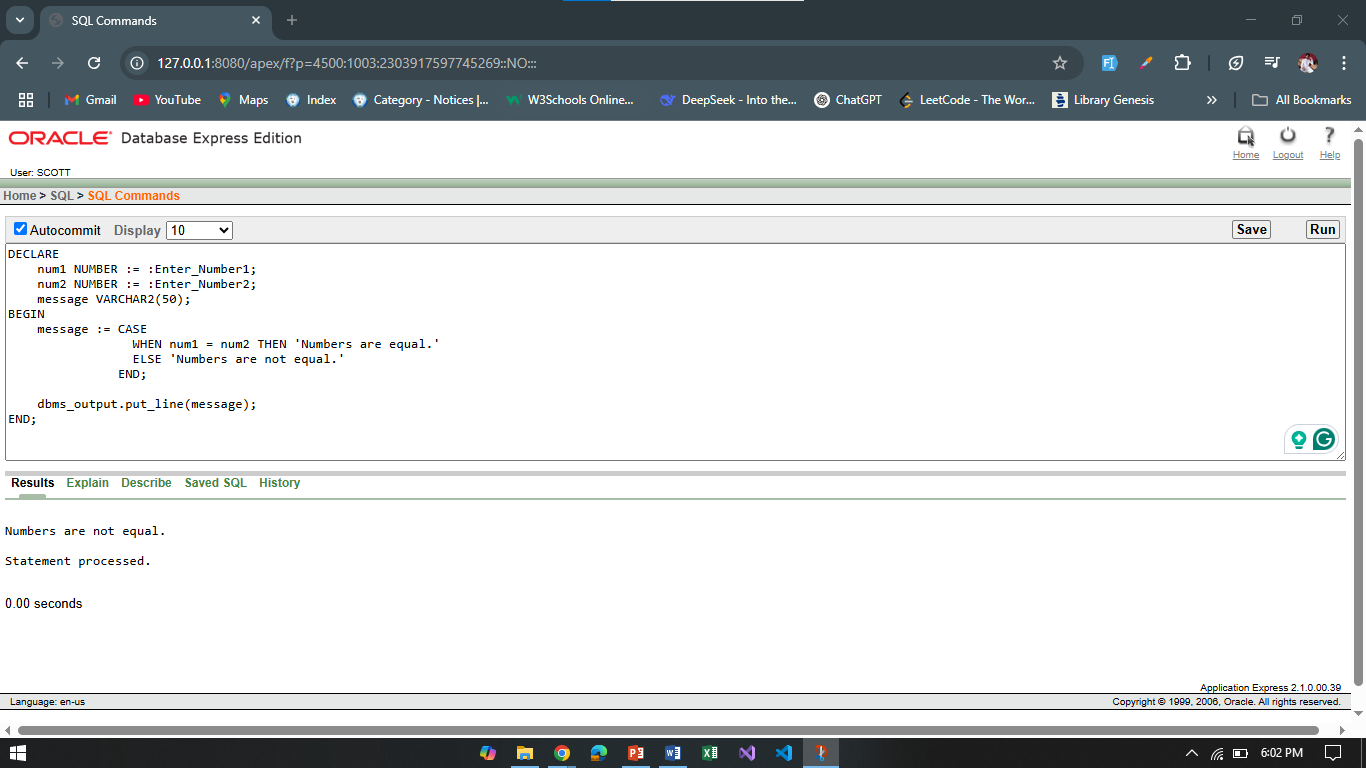
WHEN num1 = num2 THEN 'Numbers are equal.'

ELSE 'Numbers are not equal.'

END;

dbms\_output.put\_line(message);

END;



**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’.

**Answer:**

DECLARE

salary NUMBER;

BEGIN

SELECT sal INTO salary FROM emp WHERE ename = 'ALLEN';

IF salary > 2000 THEN

dbms\_output.put\_line('SALARY GREATER THAN 2000');

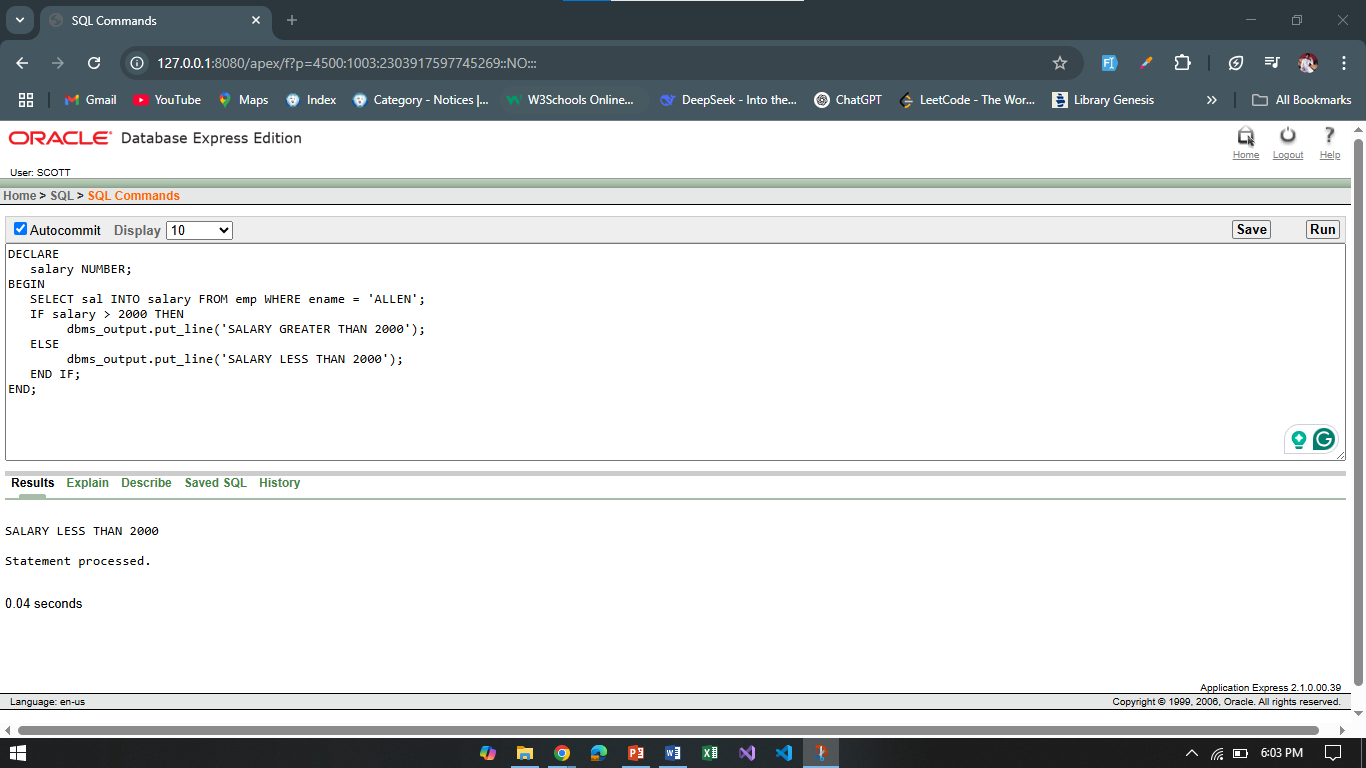
ELSE

dbms\_output.put\_line('SALARY LESS THAN 2000');

END IF;

END;

**Output:**



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

**Answer:**

DECLARE

v\_empno NUMBER;

v\_salary NUMBER;

BEGIN

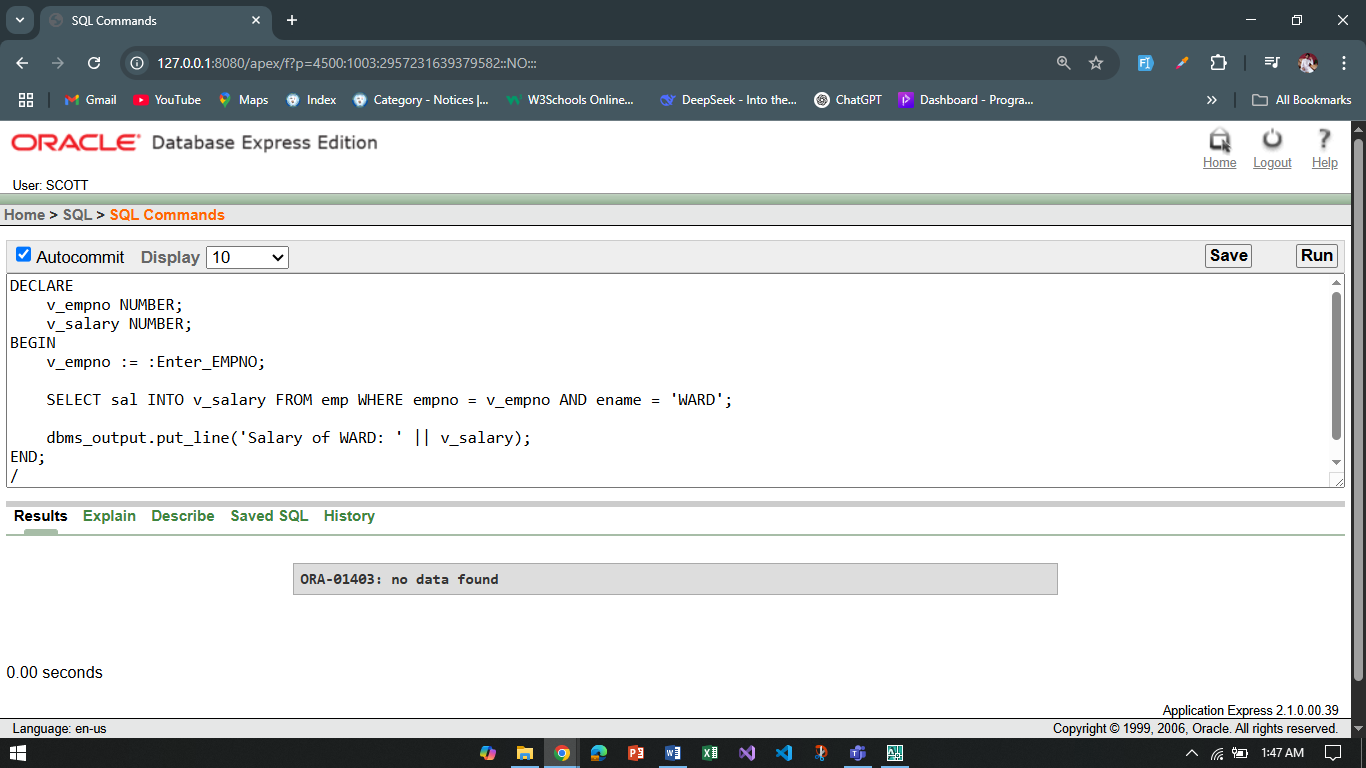
v\_empno := :Enter\_EMPNO;

SELECT sal INTO v\_salary FROM emp WHERE empno = v\_empno AND ename = 'WARD';

dbms\_output.put\_line('Salary of WARD: ' || v\_salary);

END;

/



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

**Answer:**

DECLARE

v\_empno NUMBER;

v\_salary NUMBER;

v\_ename VARCHAR2(20);

BEGIN

v\_empno := :Enter\_EMPNO;

SELECT ename, sal INTO v\_ename, v\_salary FROM emp WHERE empno = v\_empno AND ename IN ('BLAKE', 'CLARK', 'TURNER');

dbms\_output.put\_line('Employee: ' || v\_ename || ', Salary: ' || v\_salary);

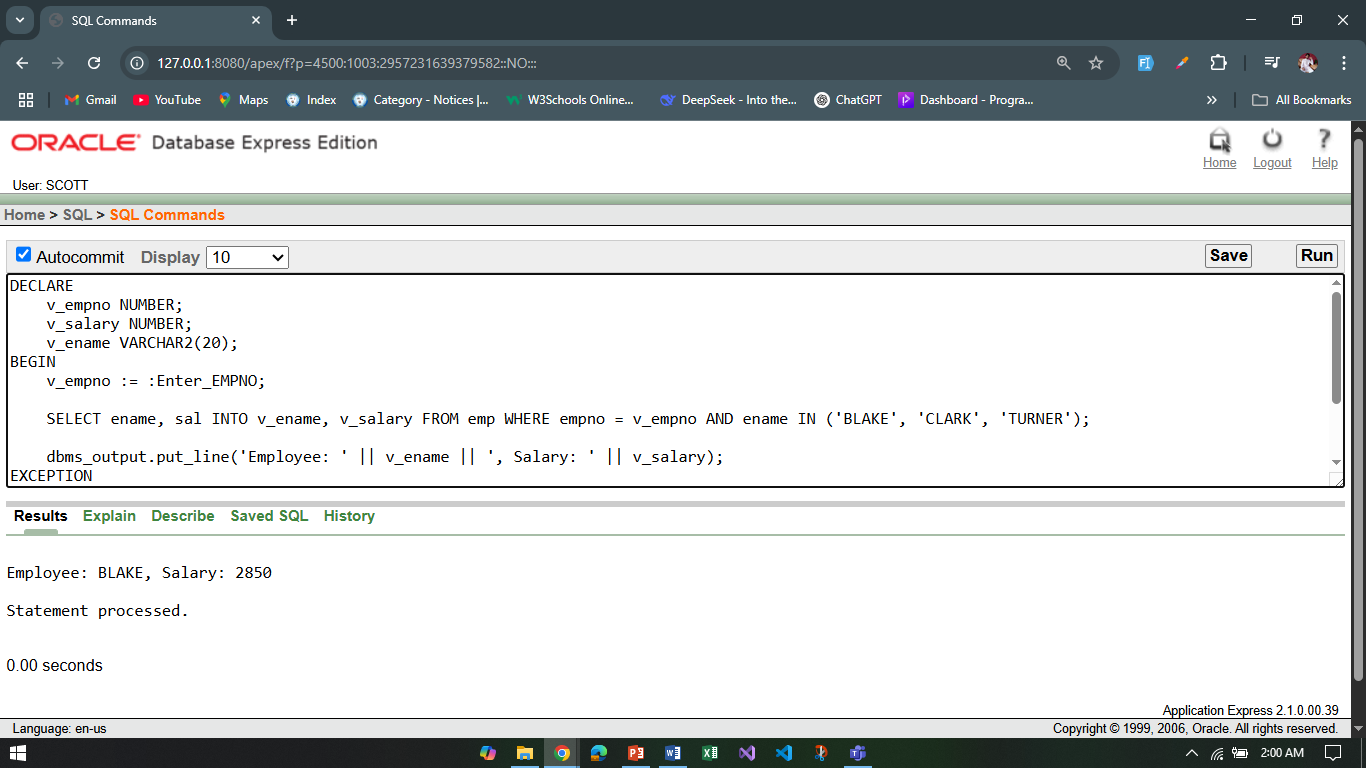
EXCEPTION

WHEN NO\_DATA\_FOUND THEN

dbms\_output.put\_line('Employee not found.');

END;

/



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.

**Answer:**

DECLARE

v\_empno NUMBER;

v\_salary NUMBER;

v\_total\_salary NUMBER := 0;

v\_ename VARCHAR2(20);

BEGIN

FOR i IN 1..3 LOOP

v\_empno := :Enter\_EMPNO;

SELECT ENAME, SAL INTO v\_ename, v\_salary

FROM EMP

WHERE EMPNO = v\_empno AND ENAME IN ('BLAKE', 'CLARK', 'TURNER');

dbms\_output.put\_line('Employee: ' || v\_ename || ', Salary: ' || v\_salary);

v\_total\_salary := v\_total\_salary + v\_salary;

END LOOP;

dbms\_output.put\_line('Total Salary: ' || v\_total\_salary);

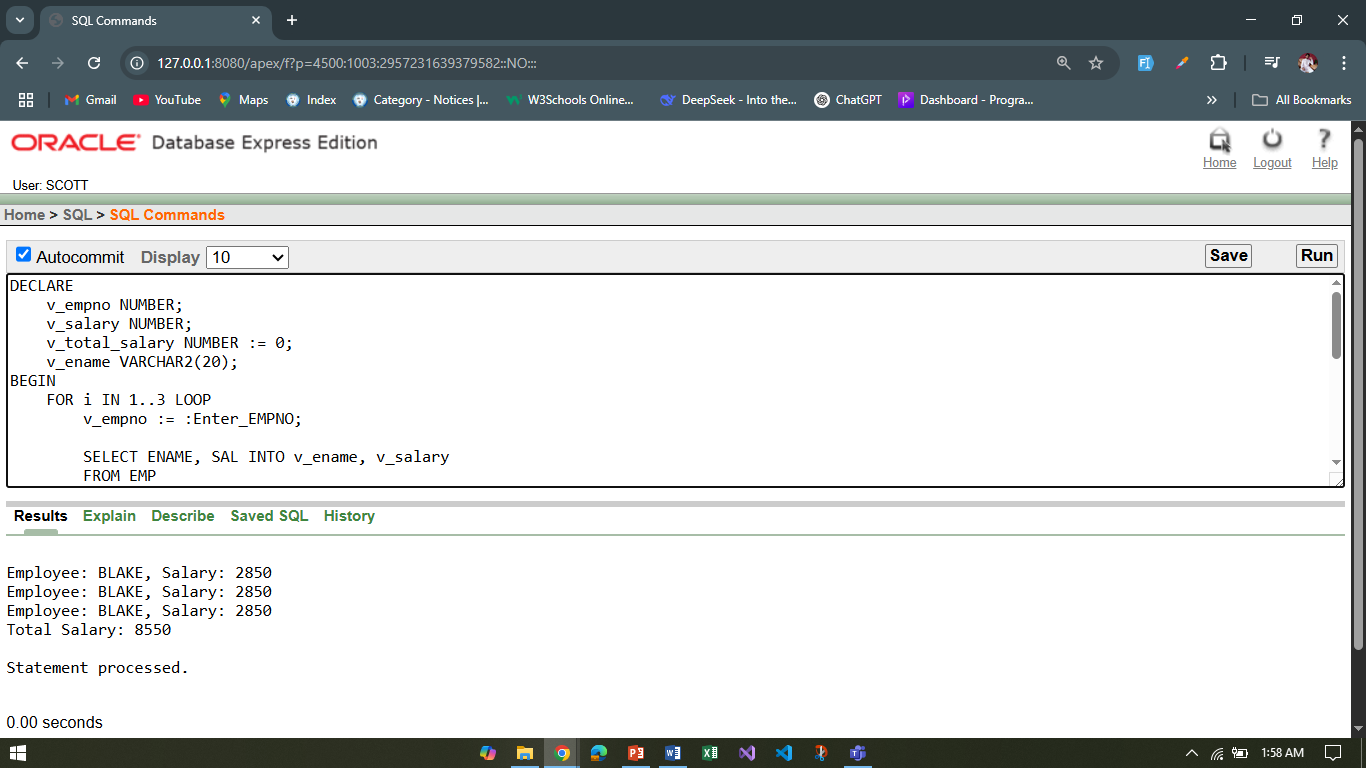
EXCEPTION

WHEN NO\_DATA\_FOUND THEN

dbms\_output.put\_line('Employee not found.');

END;

/



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

**Answer:**

DECLARE

v\_comm NUMBER;

BEGIN

SELECT comm INTO v\_comm FROM emp WHERE ename = 'SMITH';

IF v\_comm IS NULL THEN

dbms\_output.put\_line('NOT APPLICABLE FOR COMMISSION');

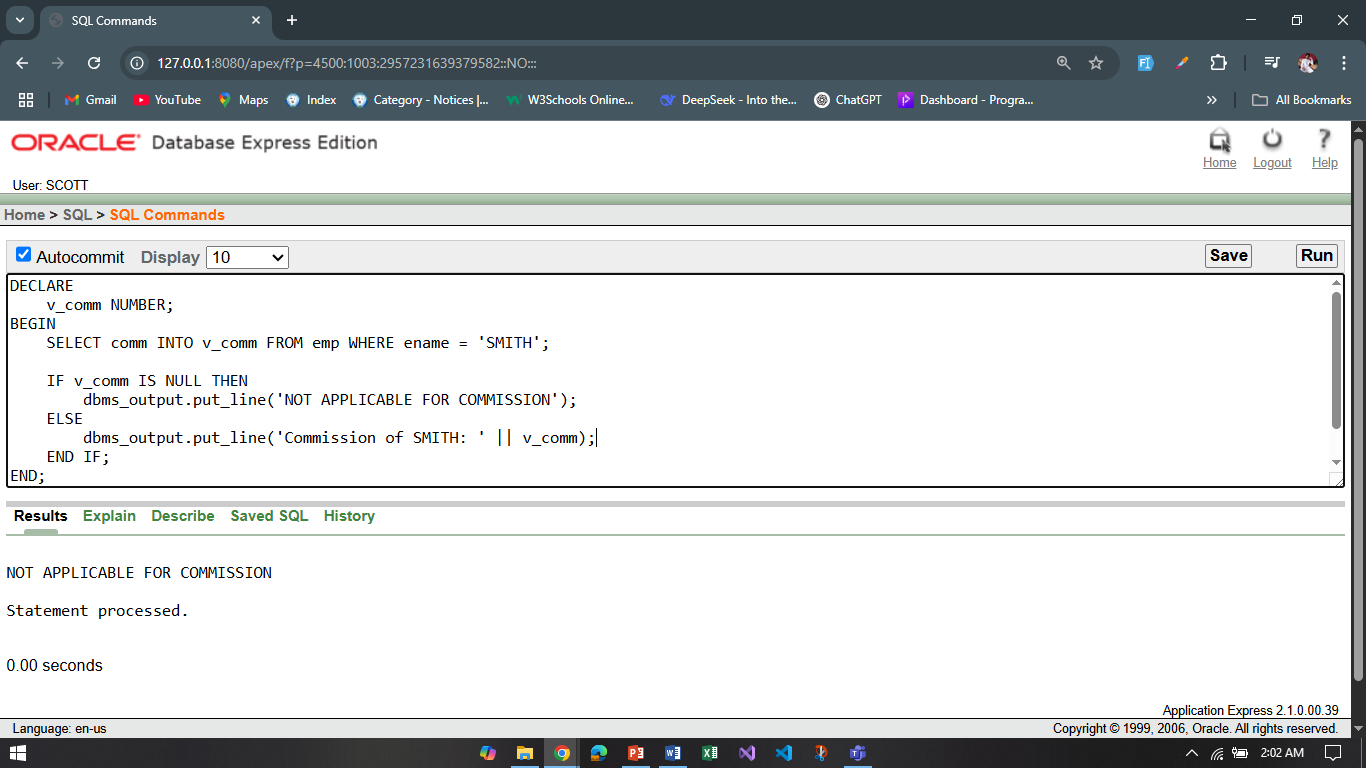
ELSE

dbms\_output.put\_line('Commission of SMITH: ' || v\_comm);

END IF;

END;

/



**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.

**Answer:**

DECLARE

cnt number := 1;

salary number;

BEGIN

LOOP

select sal into salary from emp where ename = 'JONES';

dbms\_output.put\_line(salary);

cnt := cnt + 1;

IF cnt > 3 THEN

exit;

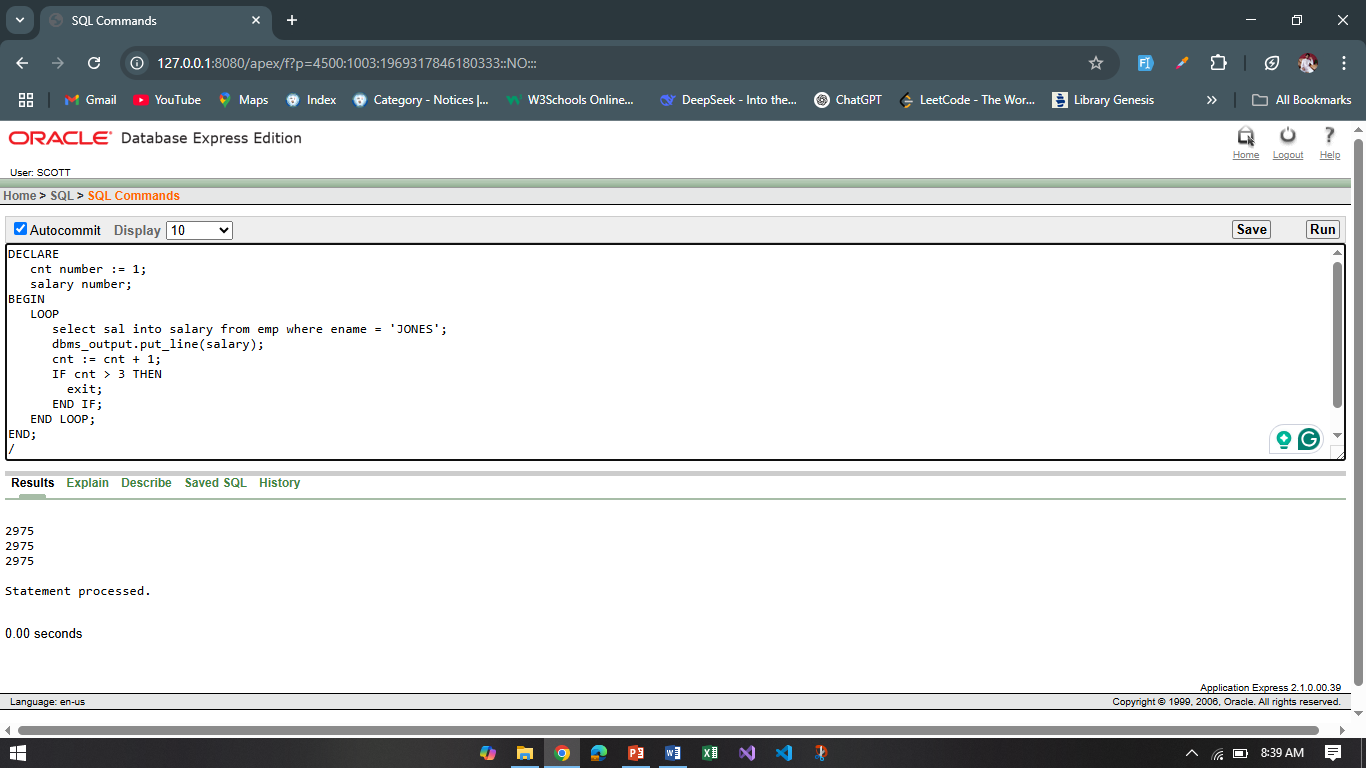
END IF;

END LOOP;

END;

/

**Output:**



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

**Answer:**

DECLARE

v\_salary NUMBER ;

counter NUMBER := 1;

BEGIN

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

WHILE counter <= 3 LOOP

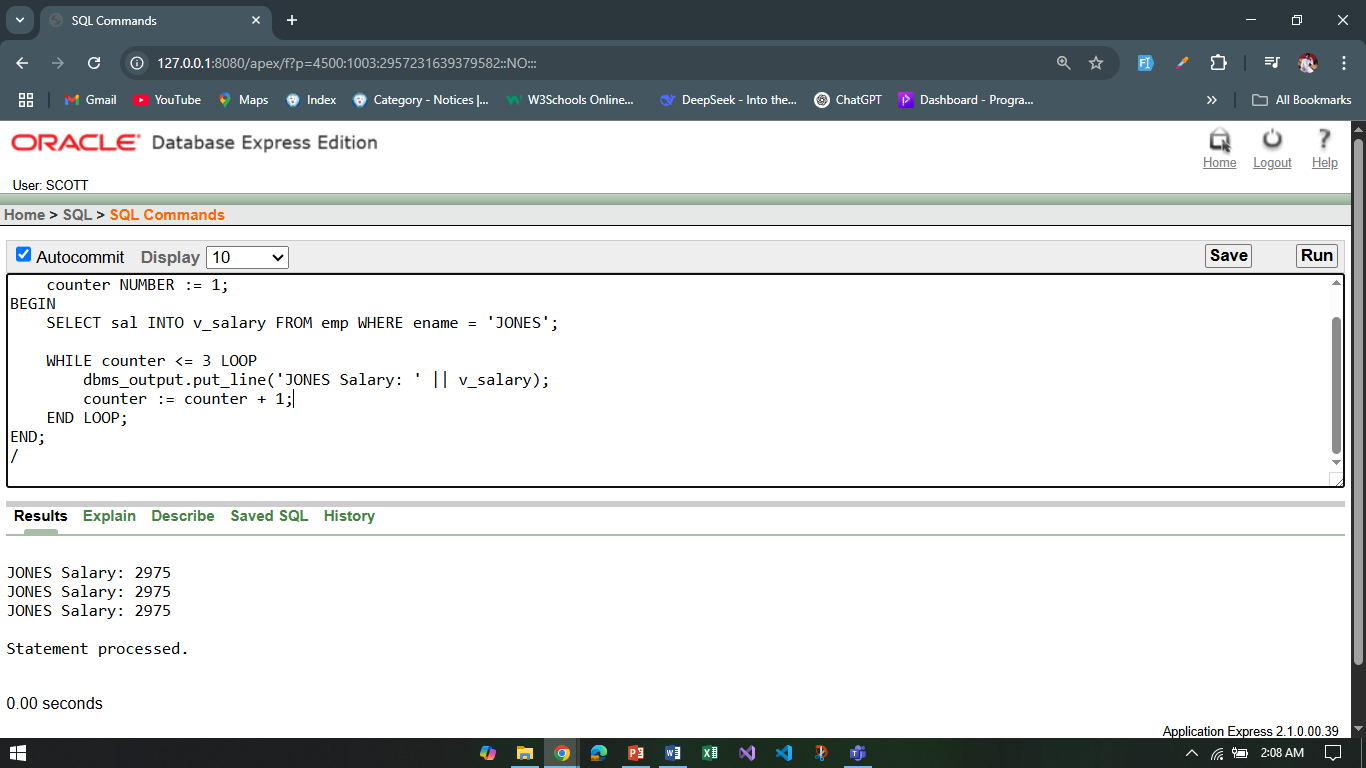
dbms\_output.put\_line('JONES Salary: ' || v\_salary);

counter := counter + 1;

END LOOP;

END;

/



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

**Answer:**

DECLARE

v\_salary number;

BEGIN

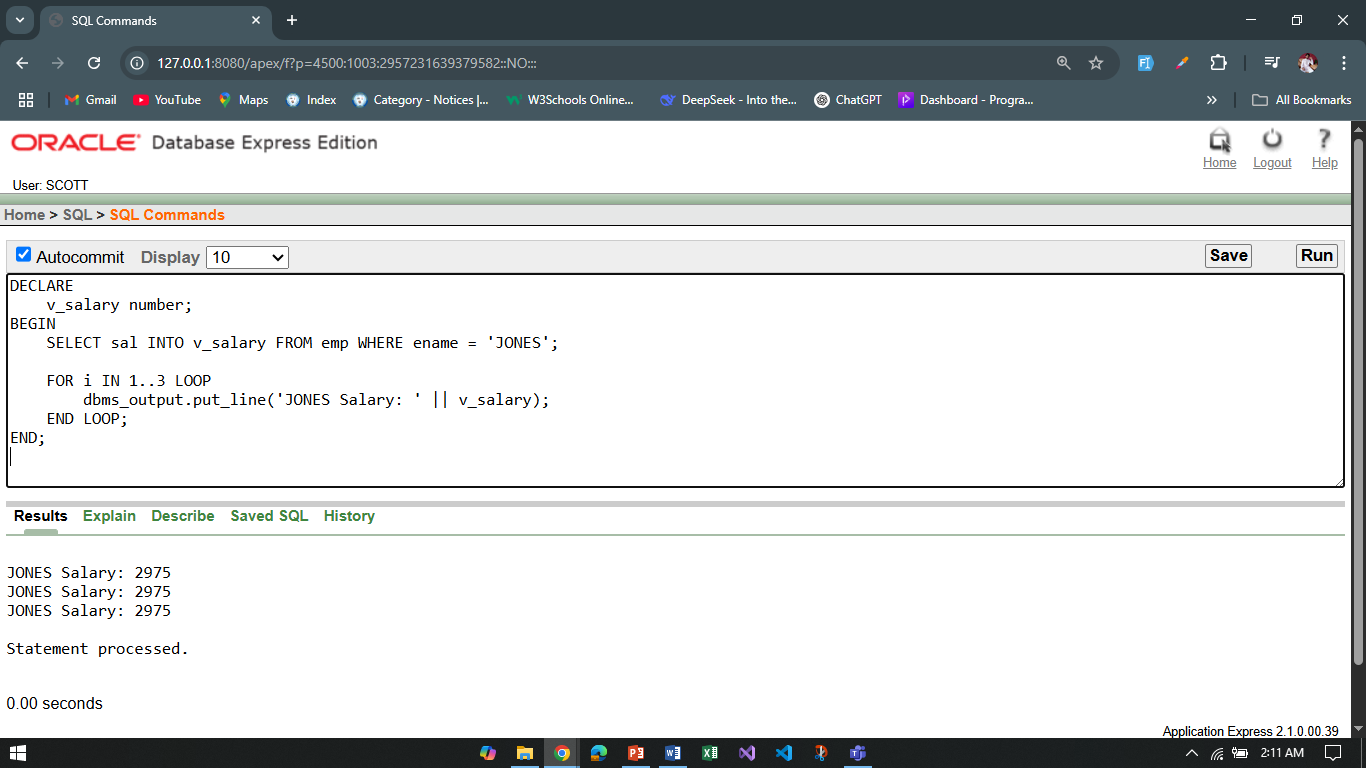
SELECT sal INTO v\_salary FROM emp WHERE ename = 'JONES';

FOR i IN 1..3 LOOP

dbms\_output.put\_line('JONES Salary: ' || v\_salary);

END LOOP;

END;



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

**Answer:**

CREATE OR REPLACE FUNCTION get\_total\_departments

RETURN NUMBER

IS

v\_total NUMBER;

BEGIN

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

RETURN v\_total;

END;

/

DECLARE

dept\_count NUMBER;

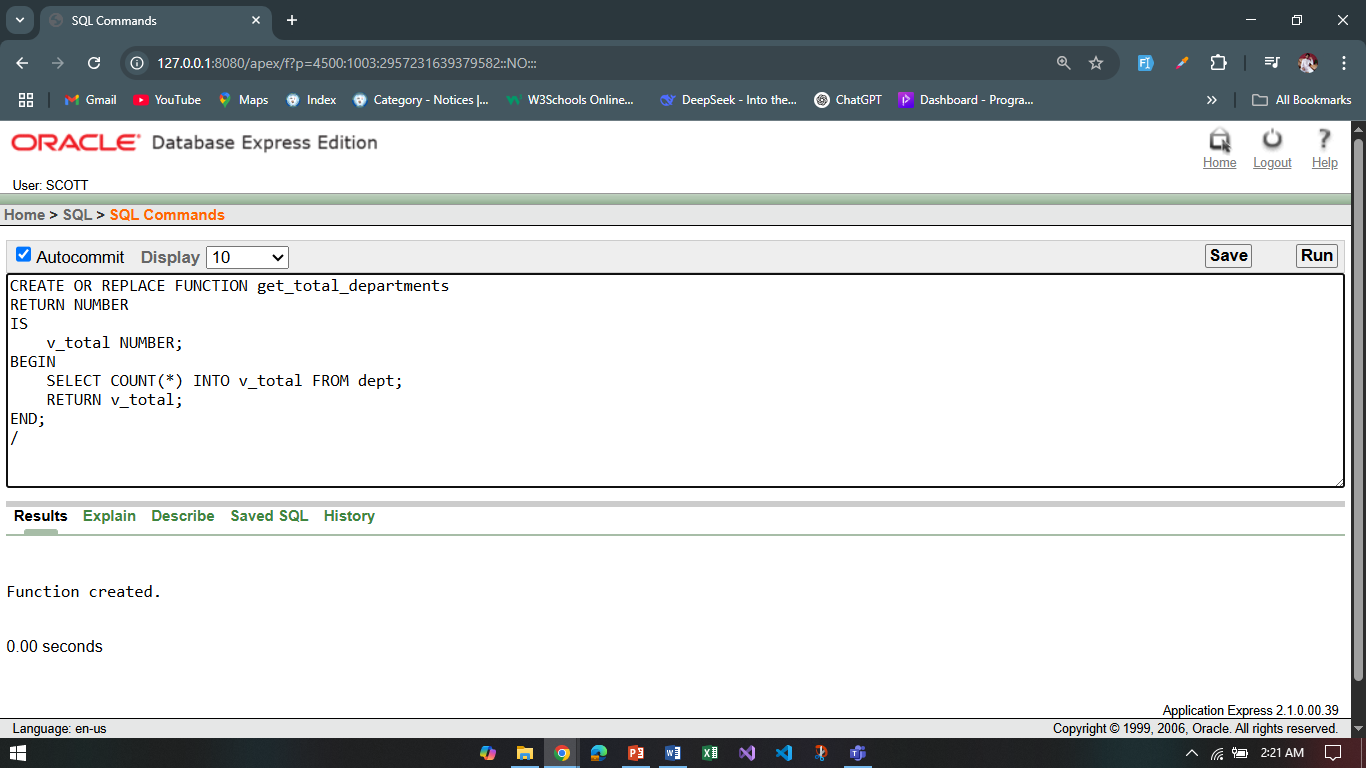
BEGIN

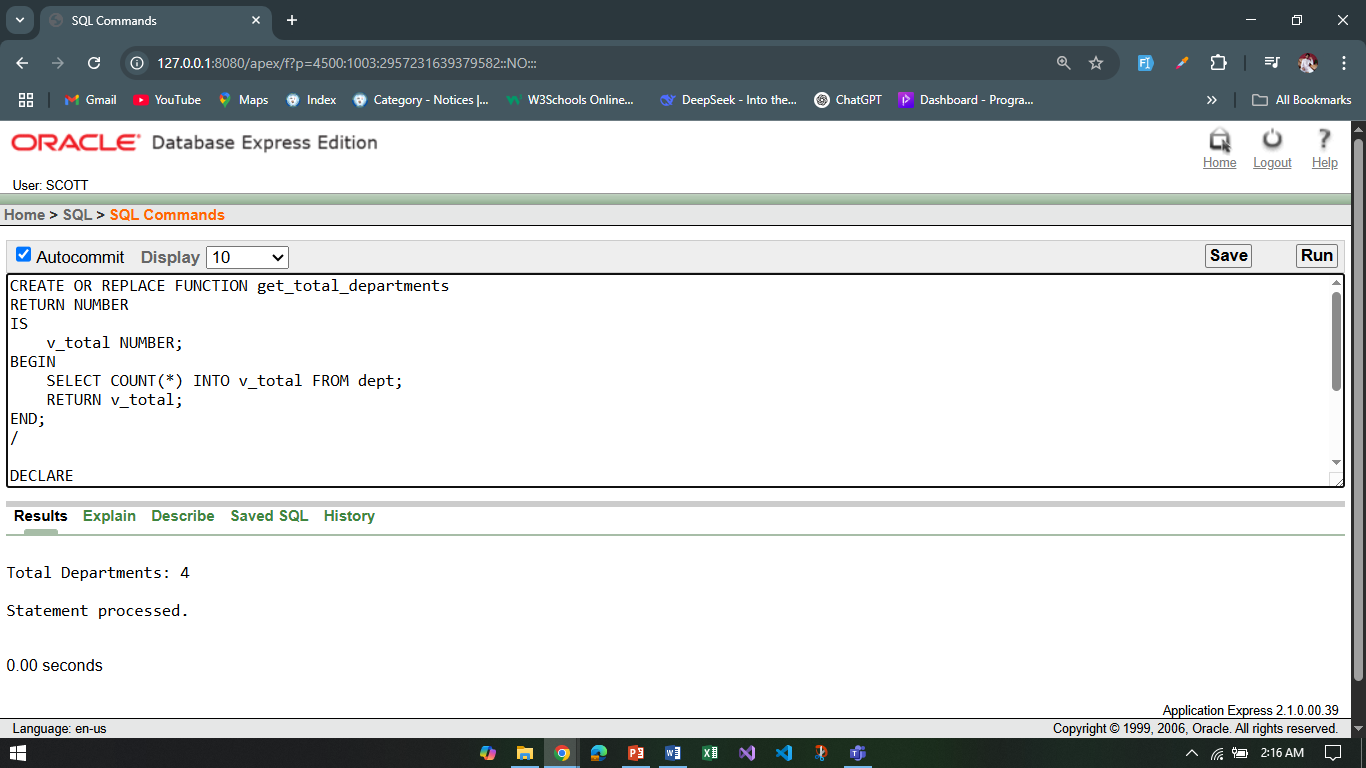
dept\_count := get\_total\_departments;

dbms\_output.put\_line('Total Departments: ' || dept\_count);

END;

/





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

**Answer:**

CREATE OR REPLACE PROCEDURE update\_allen\_salary IS

BEGIN

UPDATE emp SET sal = 100 WHERE ename = 'ALLEN';

COMMIT;

dbms\_output.put\_line('Salary of ALLEN updated to 100.');

END;

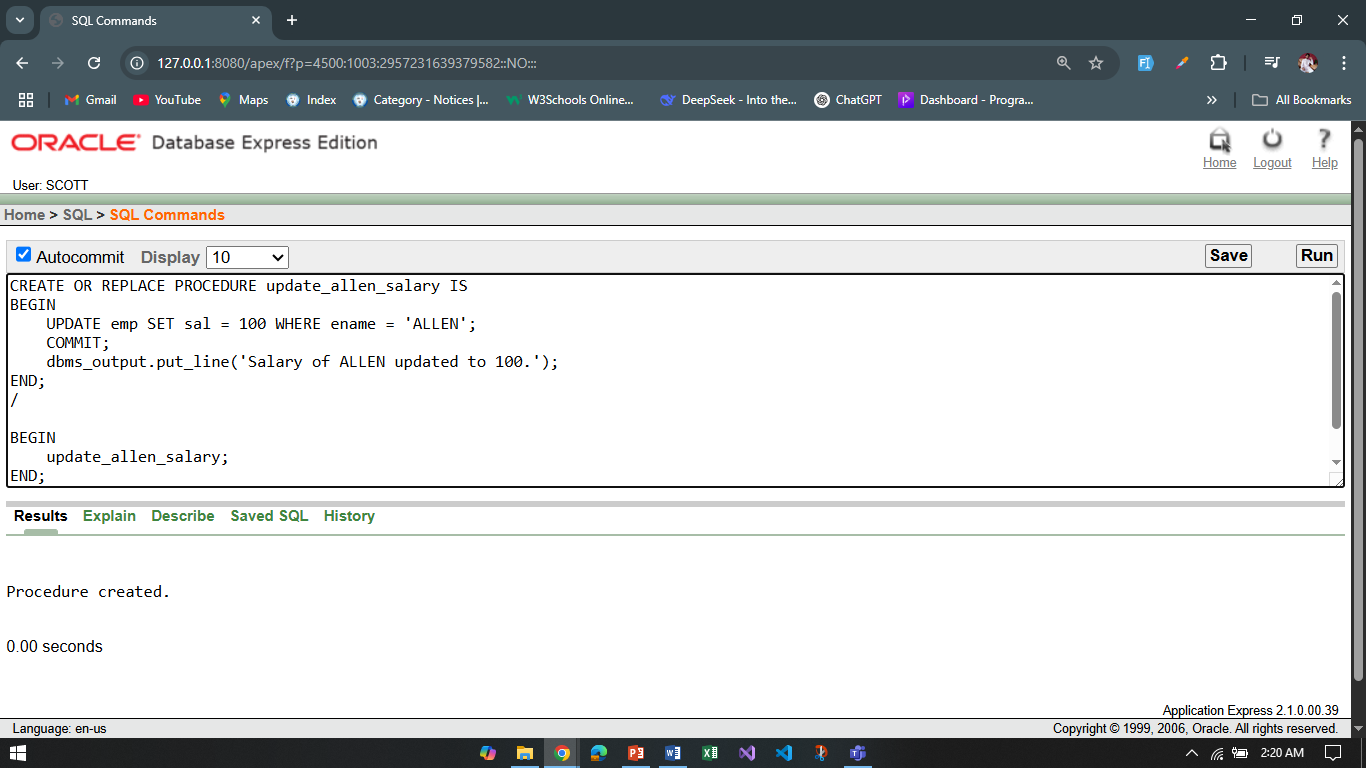
/

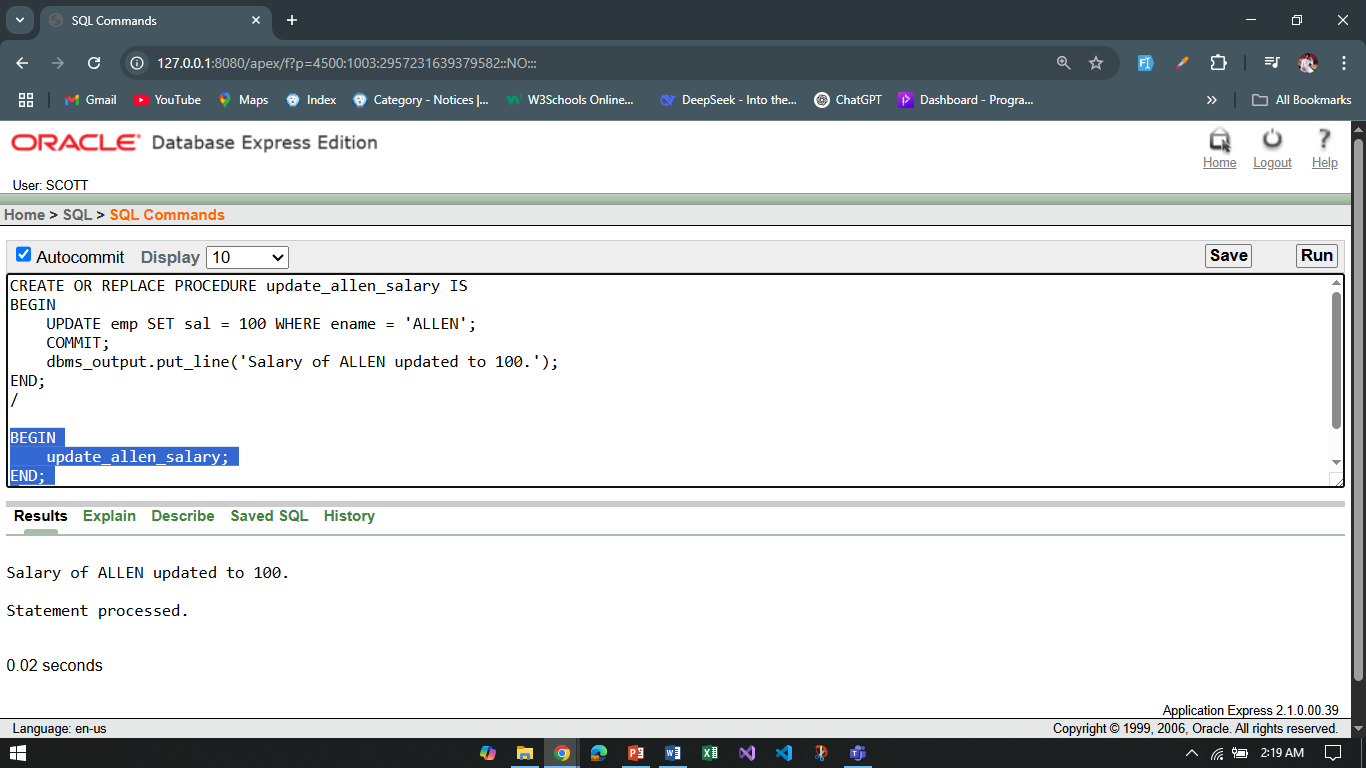
BEGIN

update\_allen\_salary;

END;

/





**\*\*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**