Khatri Likitha

21BCE11671

My solution in GitHub: [DBMS\_LabAss2](https://github.com/likitha173/DBMS_LabAss2)

***DBMS ASSIGNMENT - 2***

**PROBLEM 1:**

**Write a PL/ SQL Block to accept a year input and check whether it is a leap year.**

**Created a stored procedure in MySQL that accepts a year as input and checks whether it is a leap year.**

**CODE:**

DELIMITER $$

CREATE PROCEDURE IsLeapYear(IN year INT)

BEGIN

DECLARE leapYear BOOLEAN;

SET leapYear = FALSE;

IF year MOD 4 = 0 THEN

IF year MOD 100 = 0 THEN

IF year MOD 400 = 0 THEN

SET leapYear = TRUE;

END IF;

ELSE

SET leapYear = TRUE;

END IF;

END IF;

IF leapYear = TRUE THEN

SELECT year, ' is a leap year.' AS result;

ELSE

SELECT year, ' is not a leap year.' AS result;

END IF;

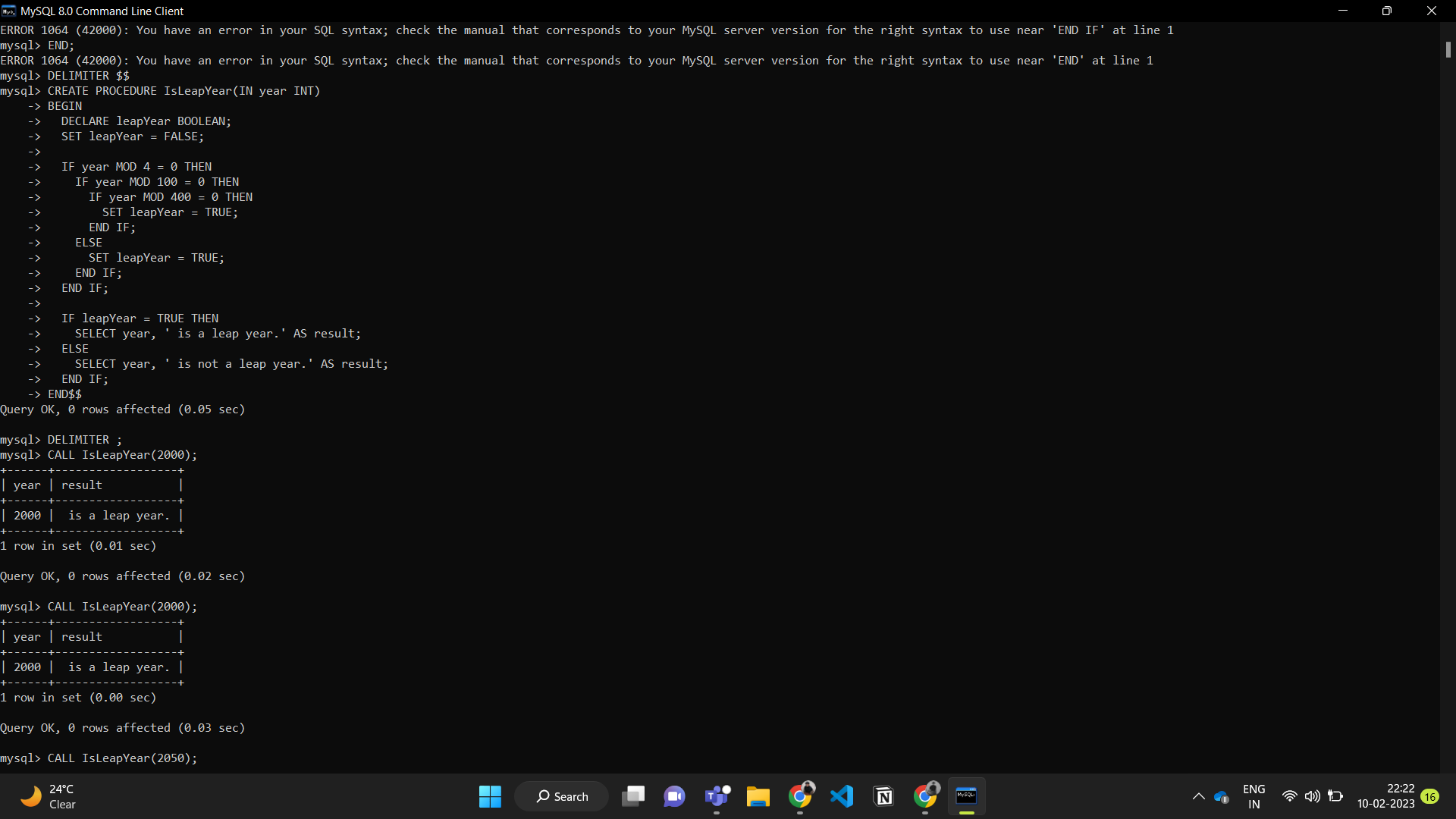
END$$

DELIMITER ;

CODE FOR CALL THE STORED PROCEDURE:

CALL IsLeapYear(2000);

**OUTPUT:**



**PROBLEM 2:**

**For the given three schemas**

**Teacher(t\_no, f\_name, l\_name, salary, supervisor, joiningdate, birthdate, title) Class(class\_no, t\_no, room\_no)**

**Payscale(Min\_limit, Max\_limit, grade)**

**(a) Calculate the bonus amount to be given to a teacher depending on the following conditions:**

**I. if salary > 10000 then bonus is 10% of the salary.**

**II. if salary is between 10000 and 20000 then bonus is 20% of the salary.**

**III. if salary is between 20000 and 25000 then bonus is 25% of the salary.**

**IV. if salary exceeds 25000 then bonus is 30% of the salary.**

**(b) Using a simple LOOP structure, list the first 10 records of the ‘teachers’ table.**

**(c) Create a procedure that selects all teachers who get a salary of Rs.20, 000 and if less than 5 teachers are getting Rs.20, 000 then give an increment of 5%. 51 Lab Manual**

**(d) Create a procedure that finds whether a teacher given by user exists or not and if not then display “teacher id not exists”.**

**(e) Using FOR loop, display name and id of all those teachers who are more than 58 years old.**

**(f) Using while loop, display details of all those teachers who are in grade ‘A’.**

**(g) Create a procedure that displays the names of all those teachers whose supervisor is ‘Suman’**

**Creating Tables And Inserting Data:**

**CODE:**

CREATE TABLE Teacher (

t\_no INT PRIMARY KEY,

f\_name VARCHAR(50),

l\_name VARCHAR(50),

salary INT,

supervisor INT,

joiningdate DATE,

birthdate DATE,

title VARCHAR(50)

);

CREATE TABLE Class (

class\_no INT PRIMARY KEY,

t\_no INT,

room\_no INT,

FOREIGN KEY (t\_no) REFERENCES Teacher(t\_no)

);

CREATE TABLE Payscale (

Min\_limit INT PRIMARY KEY,

Max\_limit INT,

grade VARCHAR(50)

);

INSERT INTO Teacher (t\_no, f\_name, l\_name, salary, supervisor, joiningdate, birthdate, title)

VALUES (1, 'John', 'Doe', 50000, 2, '2020-01-01', '1980-01-01', 'Professor'),

(2, 'Jane', 'Doe', 60000, NULL, '2019-01-01', '1981-01-01', 'Associate Professor');

INSERT INTO Class (class\_no, t\_no, room\_no)

VALUES (1, 1, 101),

(2, 2, 102);

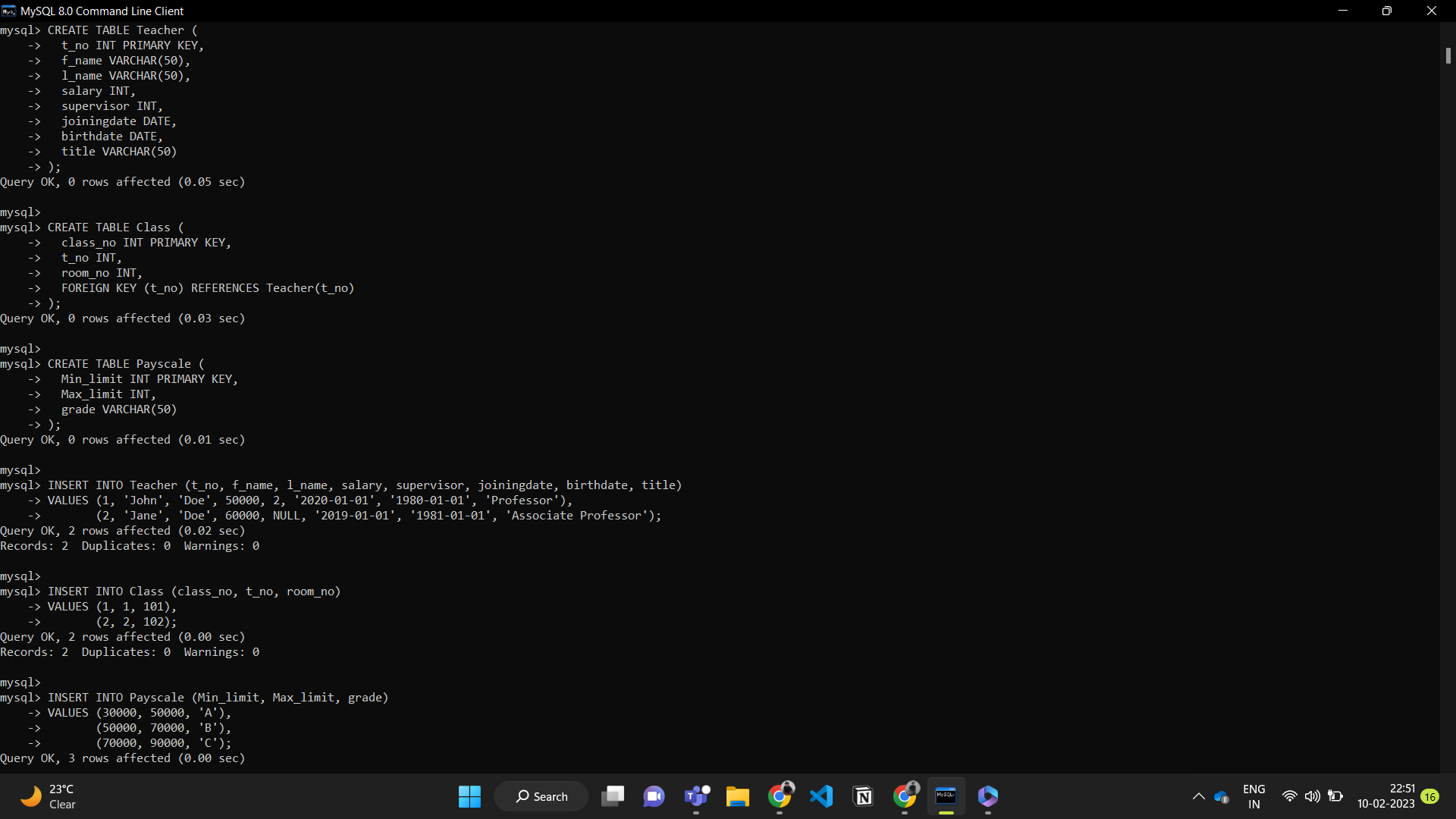
INSERT INTO Payscale (Min\_limit, Max\_limit, grade)

VALUES (30000, 50000, 'A'),

(50000, 70000, 'B'),

(70000, 90000, 'C');

**OUTPUT:**



**(a) Calculate the bonus amount to be given to a teacher depending on the following conditions:**

**I. if salary > 10000 then bonus is 10% of the salary.**

**II. if salary is between 10000 and 20000 then bonus is 20% of the salary.**

**III. if salary is between 20000 and 25000 then bonus is 25% of the salary.**

**IV. if salary exceeds 25000 then bonus is 30% of the salary.**

**CODE:**

**To calculate the bonus amount for a teacher based on the conditions**

SELECT t\_no, f\_name, l\_name, salary,

CASE

WHEN salary > 25000 THEN salary \* 0.3

WHEN salary BETWEEN 20000 AND 25000 THEN salary \* 0.25

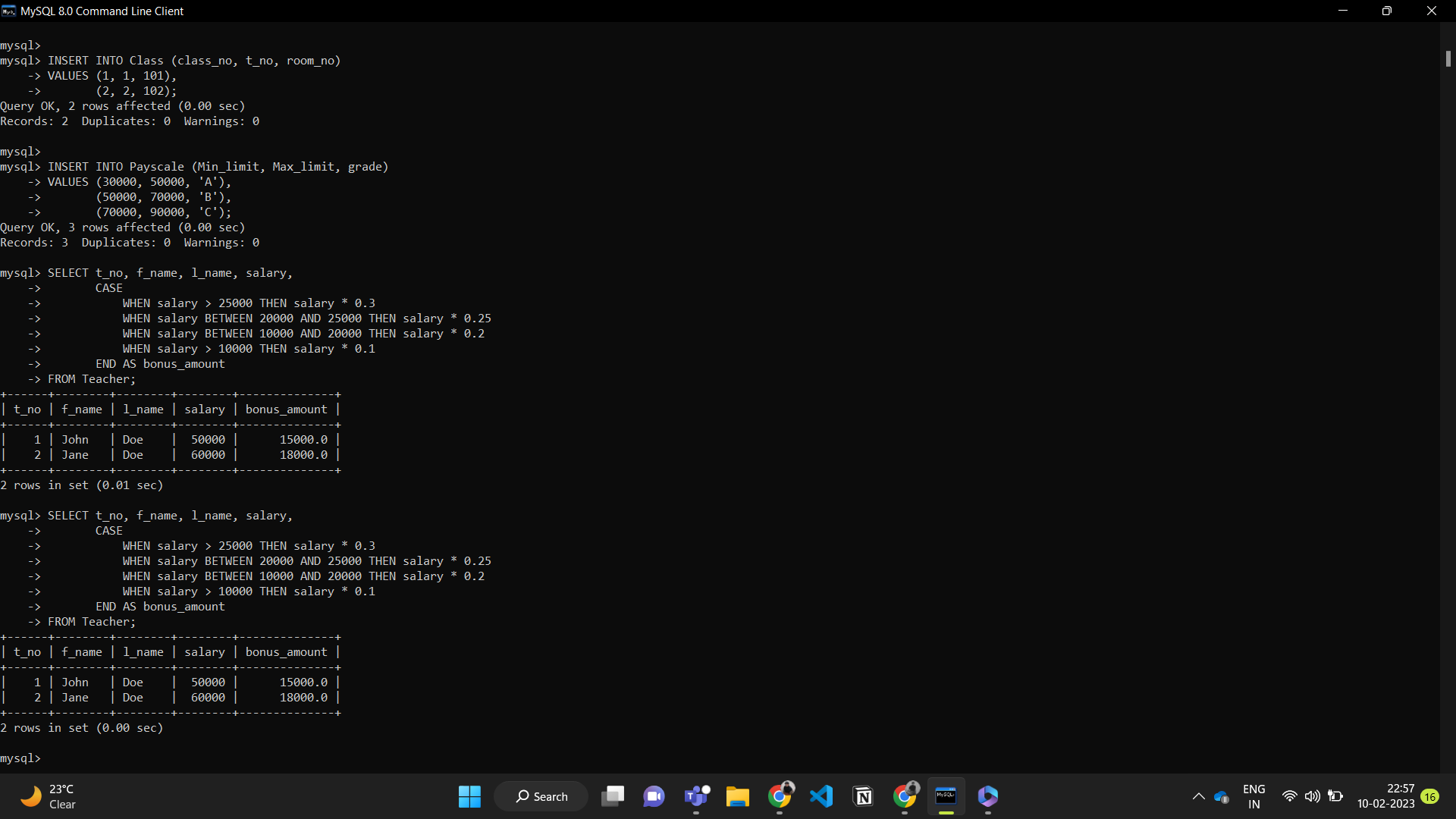
WHEN salary BETWEEN 10000 AND 20000 THEN salary \* 0.2

WHEN salary > 10000 THEN salary \* 0.1

END AS bonus\_amount

FROM Teacher;

**OUTPUT:**



**(b) Using a simple LOOP structure, list the first 10 records of the ‘teachers’ table.**

**CODE:**

**To list the first 10 records of the teachers table using a simple loop structure.**

DELIMITER $$

CREATE PROCEDURE list\_teachers()

BEGIN

DECLARE counter INT DEFAULT 0;

SELECT t\_no, f\_name, l\_name, salary INTO @t\_no, @f\_name, @l\_name, @salary

FROM Teacher

LIMIT 1;

WHILE counter < 10 DO

SELECT @t\_no, @f\_name, @l\_name, @salary;

SET counter = counter + 1;

SELECT t\_no, f\_name, l\_name, salary INTO @t\_no, @f\_name, @l\_name, @salary

FROM Teacher

WHERE t\_no > @t\_no

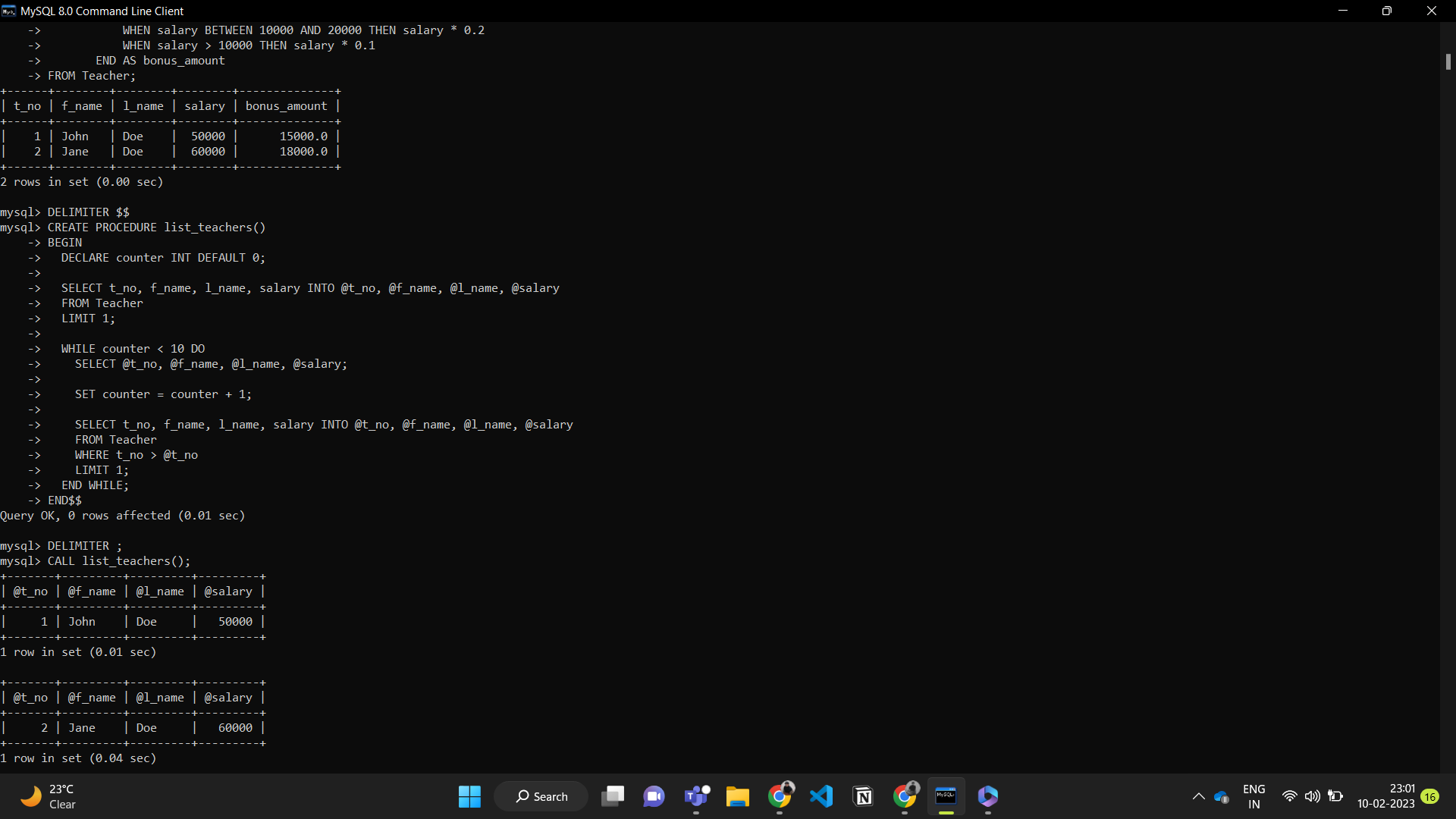
LIMIT 1;

END WHILE;

END$$

DELIMITER ;

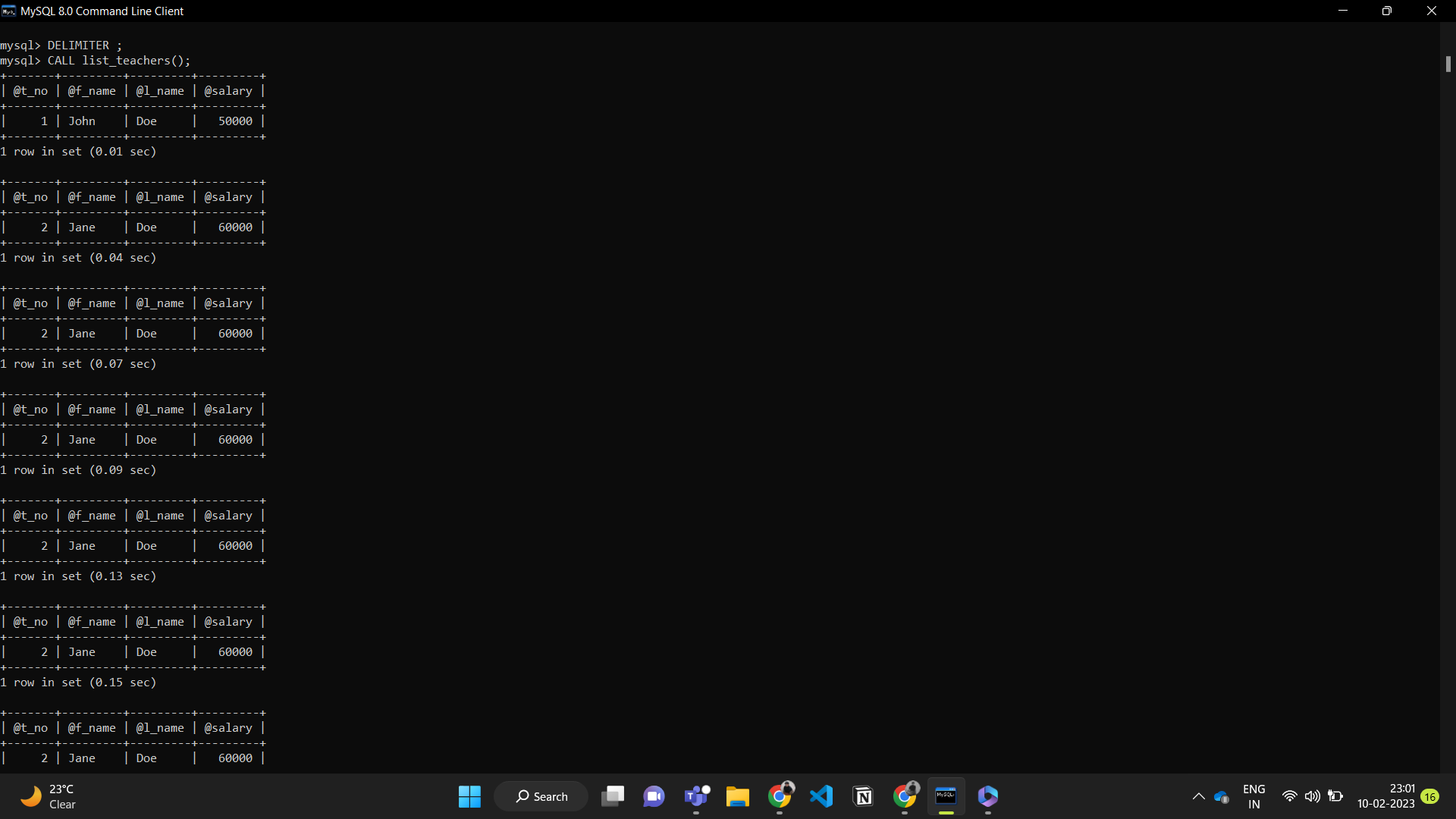
**OUTPUT:**



CODE FOR CALL THE PROCEDURE:

CALL list\_teachers();

**OUTPUT:**



**(c) Create a procedure that selects all teachers who get a salary of Rs.20, 000 and if less than 5 teachers are getting Rs.20, 000 then give an increment of 5%. 51 Lab Manual**

**CODE:**

**To create a procedure in MySQL that selects all teachers who receive a salary of Rs. 20,000 and gives a 5% salary increase if less than 5 teachers receive this salary:**

DELIMITER $$

CREATE PROCEDURE give\_increment()

BEGIN

DECLARE total\_teachers INT DEFAULT 0;

SELECT COUNT(\*) INTO total\_teachers

FROM Teacher

WHERE salary = 20000;

IF total\_teachers < 5 THEN

UPDATE Teacher

SET salary = salary \* 1.05

WHERE salary = 20000;

END IF;

SELECT \*

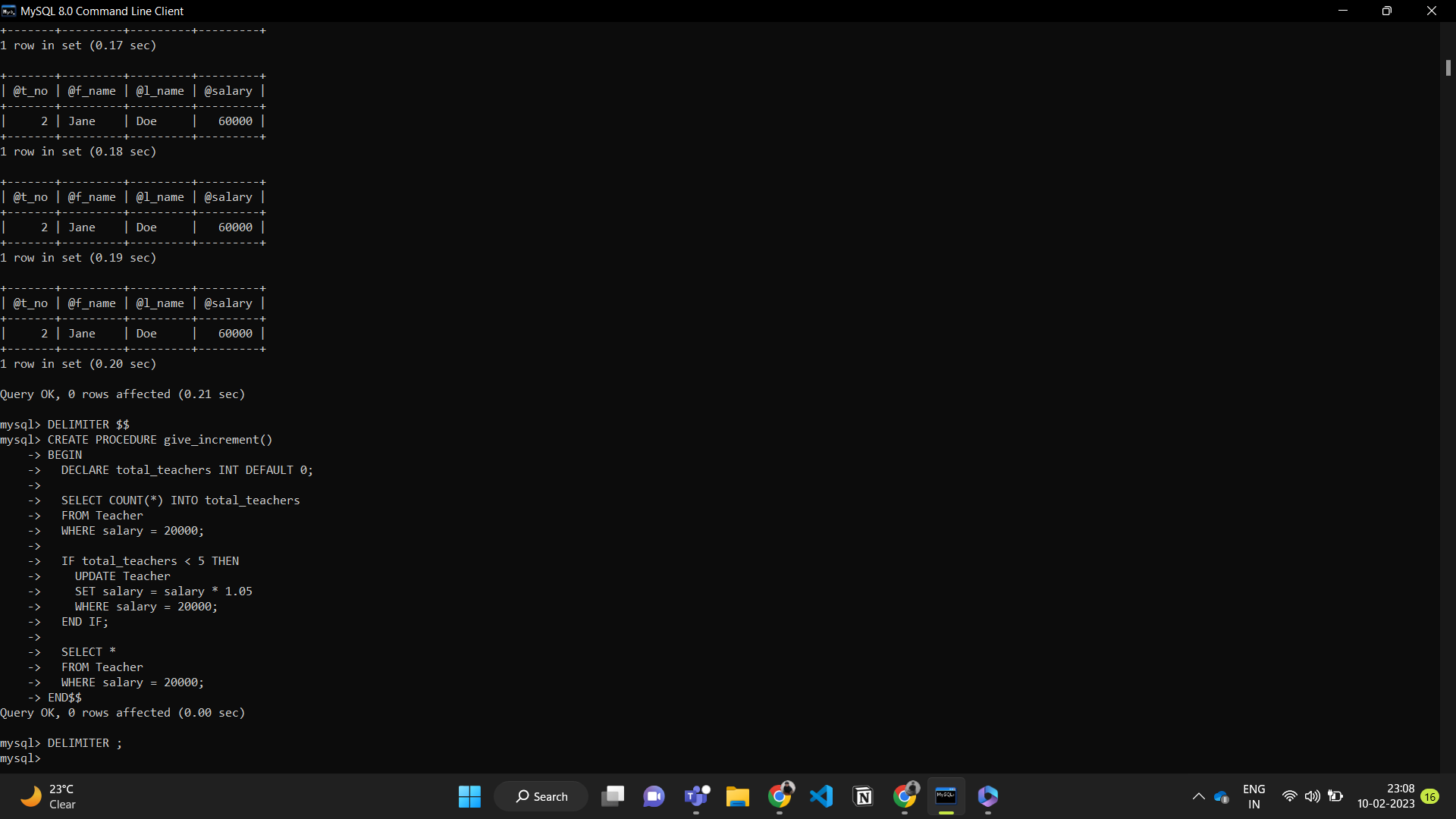
FROM Teacher

WHERE salary = 20000;

END$$

DELIMITER ;

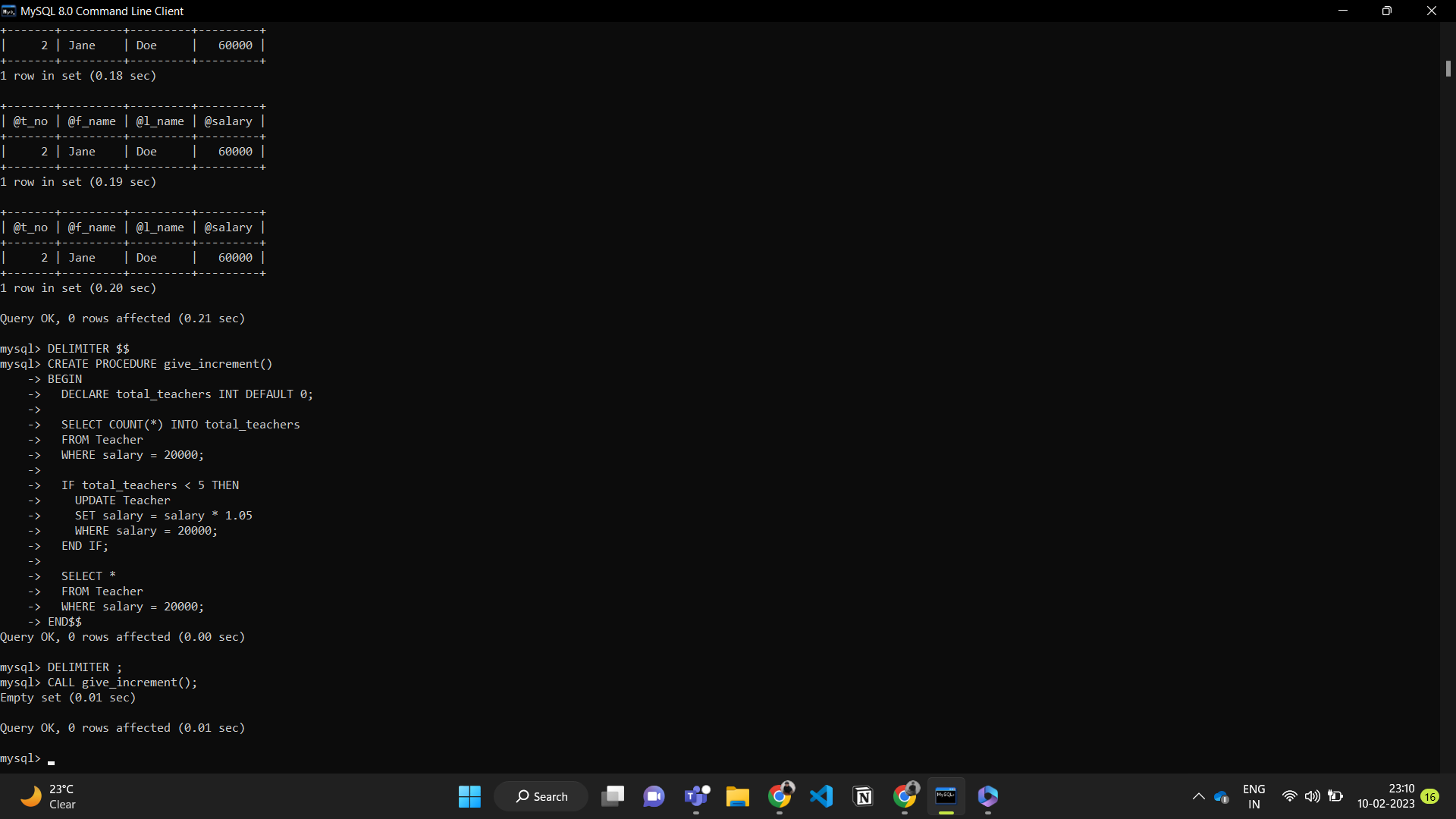
**OUTPUT:**



CODE FOR CALL THE PROCEDURE:

CALL give\_increment();

**OUTPUT:**



**d) Create a procedure that finds whether a teacher given by user exists or not and if not then display “teacher id not exists”.**

**CODE:**

**To create a procedure in MySQL that finds whether a teacher given by the user exists or not and displays a message if the teacher doesn't exist:**

DELIMITER $$

CREATE PROCEDURE find\_teacher(IN teacher\_id INT)

BEGIN

DECLARE teacher\_exists INT DEFAULT 0;

SELECT COUNT(\*) INTO teacher\_exists

FROM Teacher

WHERE t\_no = teacher\_id;

IF teacher\_exists = 0 THEN

SELECT 'Teacher ID not exists';

ELSE

SELECT \*

FROM Teacher

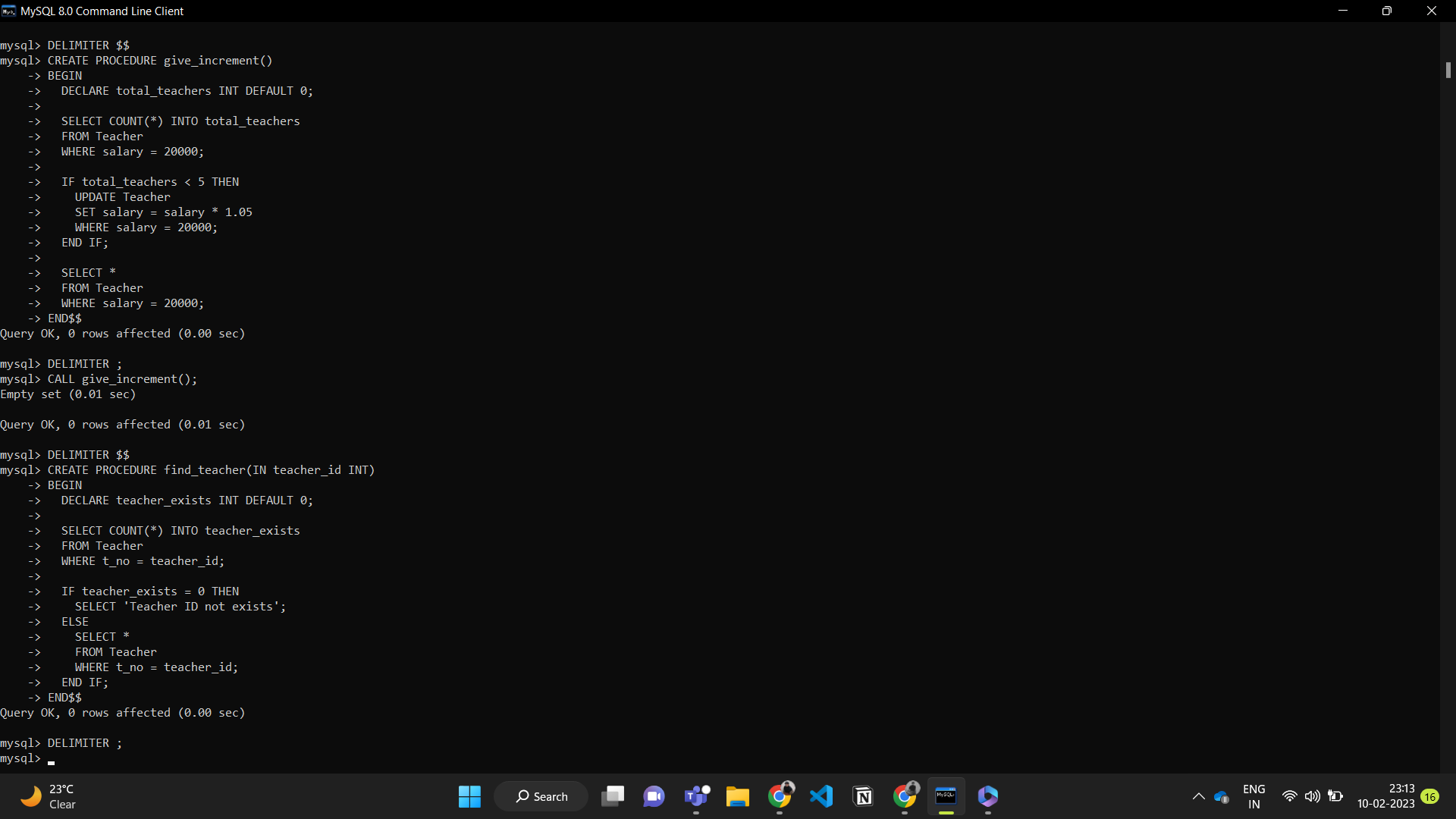
WHERE t\_no = teacher\_id;

END IF;

END$$

DELIMITER ;

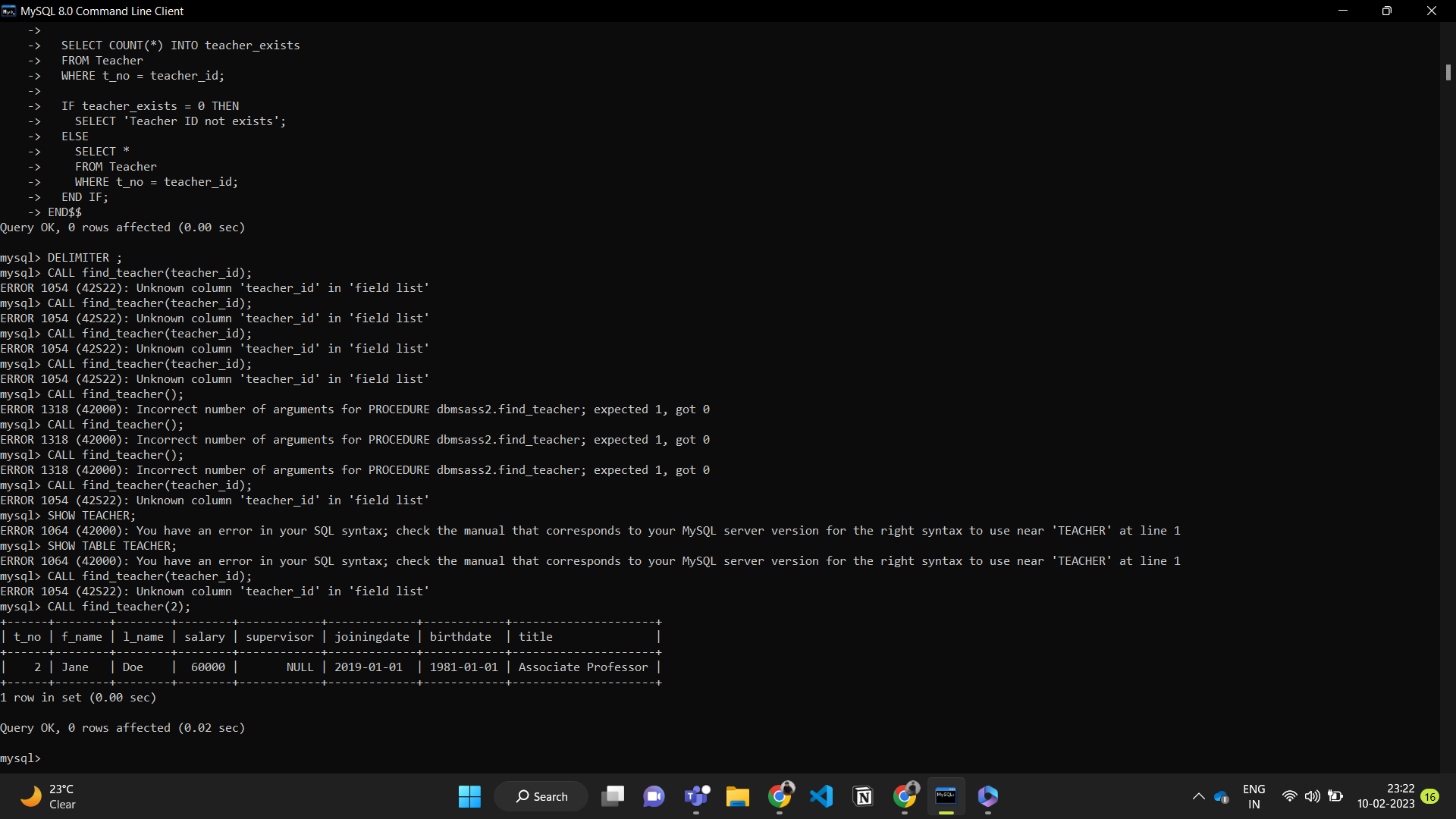
**OUTPUT:**



CODE FOR CALL THE PROCEDURE:

CALL find\_teacher(2);

**OUTPUT:**



**(e) Using FOR loop, display name and id of all those teachers who are more than 58 years old.**

**CODE:**

**To display the name and ID of all teachers who are more than 58 years old using a FOR loop in MySQL:**

DELIMITER $$

CREATE PROCEDURE find\_old\_teachers()

BEGIN

DECLARE teacher\_birthdate DATE;

DECLARE teacher\_id INT;

DECLARE teacher\_name VARCHAR(100);

DECLARE teacher\_age INT;

DECLARE done INT DEFAULT 0;

DECLARE cur CURSOR FOR

SELECT t\_no, CONCAT(f\_name, ' ', l\_name) AS name, birthdate

FROM Teacher;

DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;

OPEN cur;

REPEAT

FETCH cur INTO teacher\_id, teacher\_name, teacher\_birthdate;

IF NOT done THEN

SET teacher\_age = YEAR(CURDATE()) - YEAR(teacher\_birthdate);

IF teacher\_age > 58 THEN

SELECT teacher\_id, teacher\_name;

END IF;

END IF;

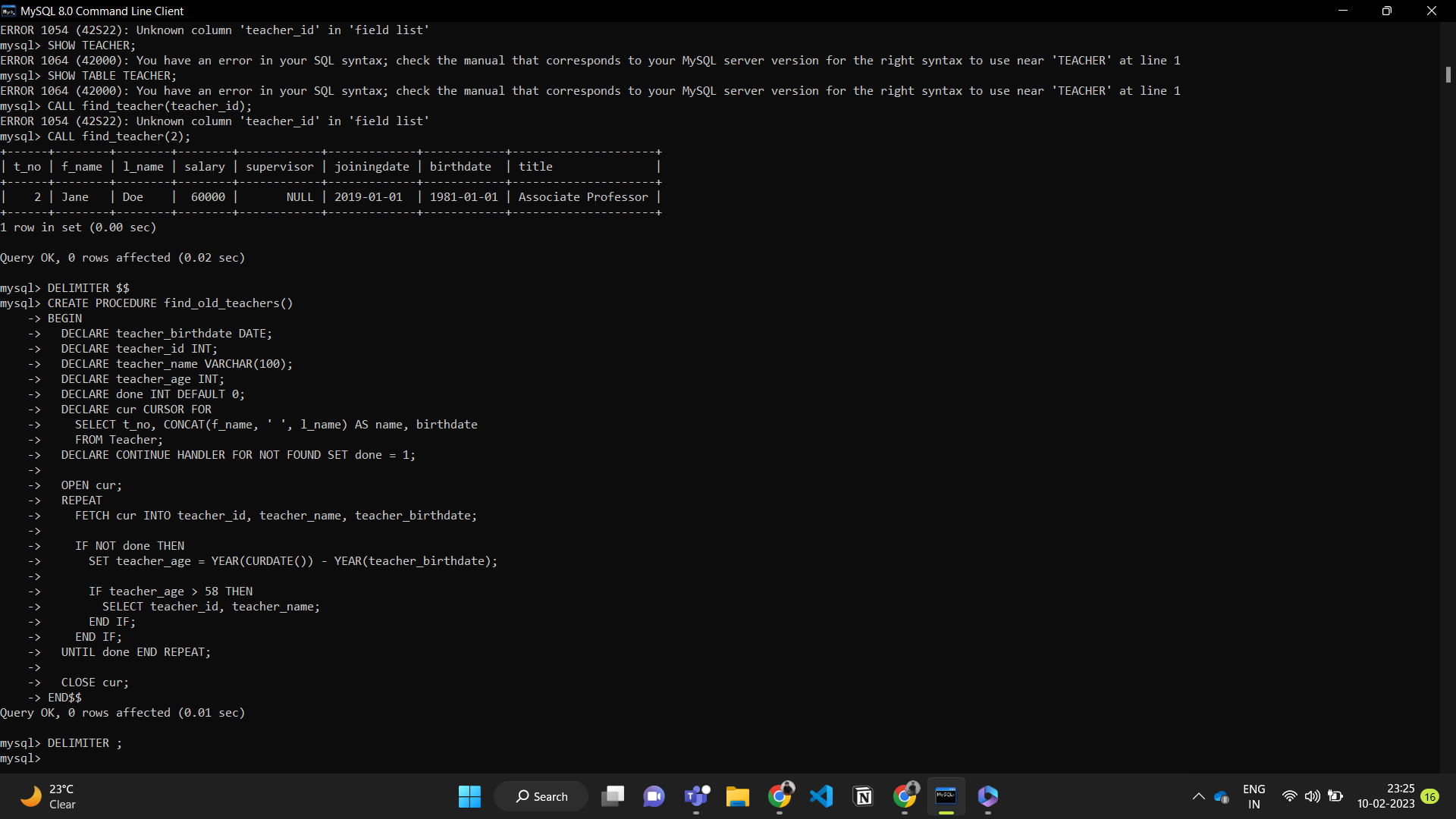
UNTIL done END REPEAT;

CLOSE cur;

END$$

DELIMITER ;

**OUTPUT:**



**(f) Using while loop, display details of all those teachers who are in grade ‘A’.**

**CODE:**

**To display the details of all teachers who are in grade 'A' using a WHILE loop in MySQL:**

DELIMITER $$

CREATE PROCEDURE find\_teachers\_in\_grade\_A()

BEGIN

DECLARE teacher\_id INT;

DECLARE teacher\_name VARCHAR(100);

DECLARE teacher\_salary INT;

DECLARE teacher\_grade CHAR(1);

DECLARE done INT DEFAULT 0;

DECLARE cur CURSOR FOR

SELECT t.t\_no, CONCAT(t.f\_name, ' ', t.l\_name) AS name, t.salary, p.grade

FROM Teacher t

INNER JOIN Payscale p ON t.salary BETWEEN p.Min\_limit AND p.Max\_limit;

DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;

OPEN cur;

REPEAT

FETCH cur INTO teacher\_id, teacher\_name, teacher\_salary, teacher\_grade;

IF NOT done THEN

IF teacher\_grade = 'A' THEN

SELECT teacher\_id, teacher\_name, teacher\_salary, teacher\_grade;

END IF;

END IF;

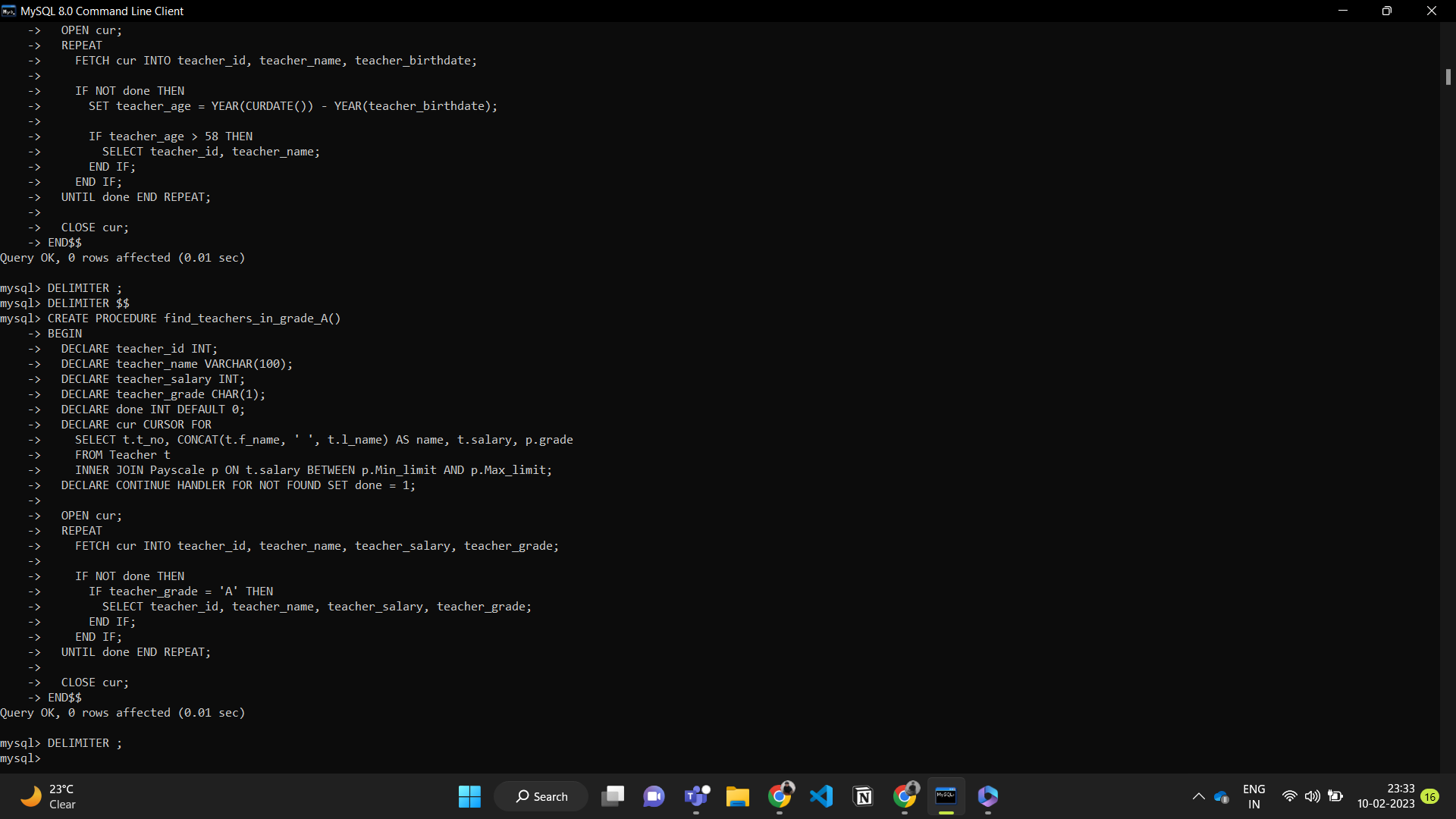
UNTIL done END REPEAT;

CLOSE cur;

END$$

DELIMITER ;

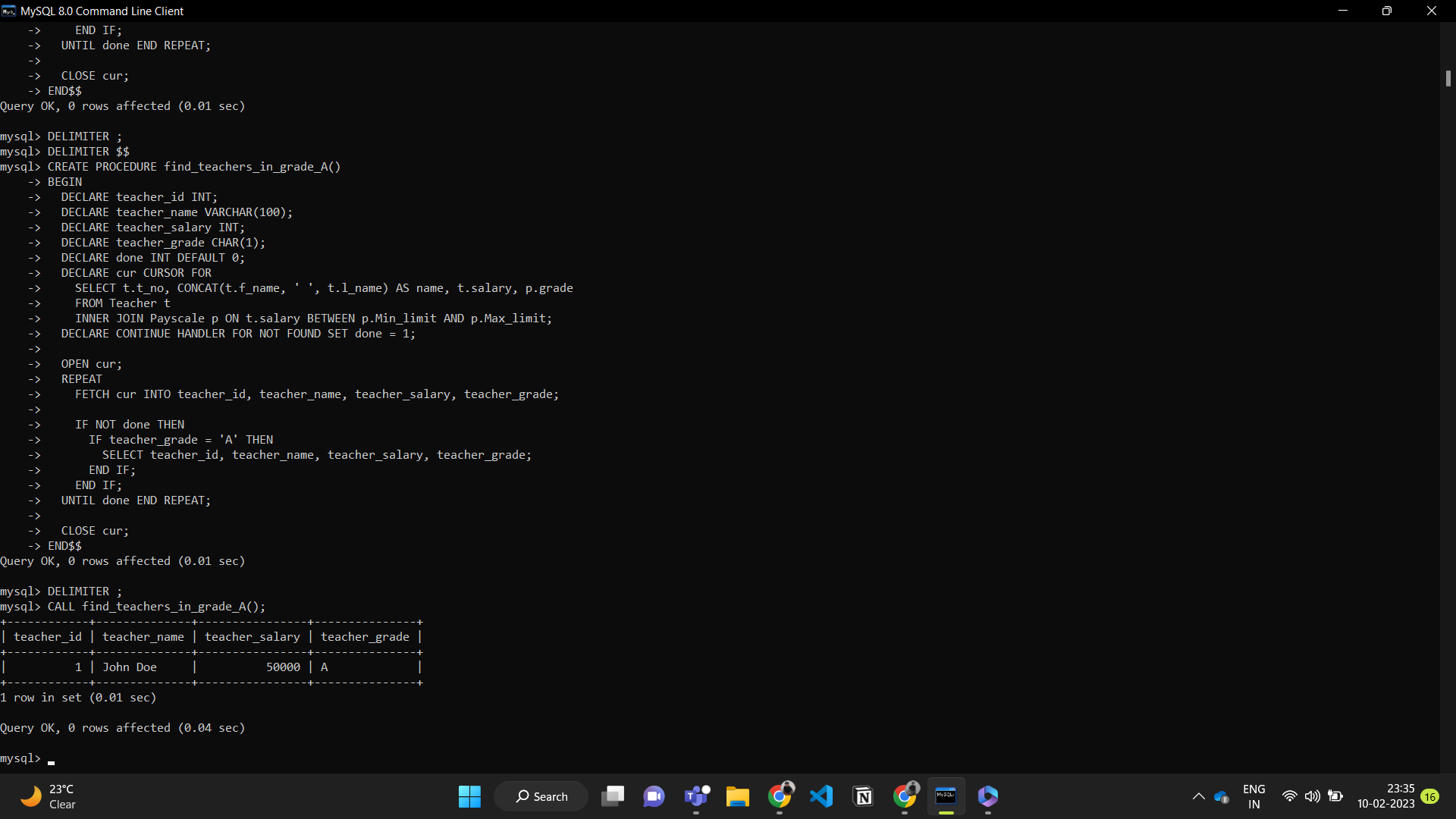
**OUTPUT:**



CODE FOR CALL THE PROCEDURE:

CALL find\_teachers\_in\_grade\_A();

**OUTPUT:**



**(g) Create a procedure that displays the names of all those teachers whose supervisor is ‘Suman’.**

**CODE:**

DELIMITER $$

CREATE PROCEDURE find\_teachers\_under\_suman()

BEGIN

SELECT CONCAT(f\_name, ' ', l\_name) AS name

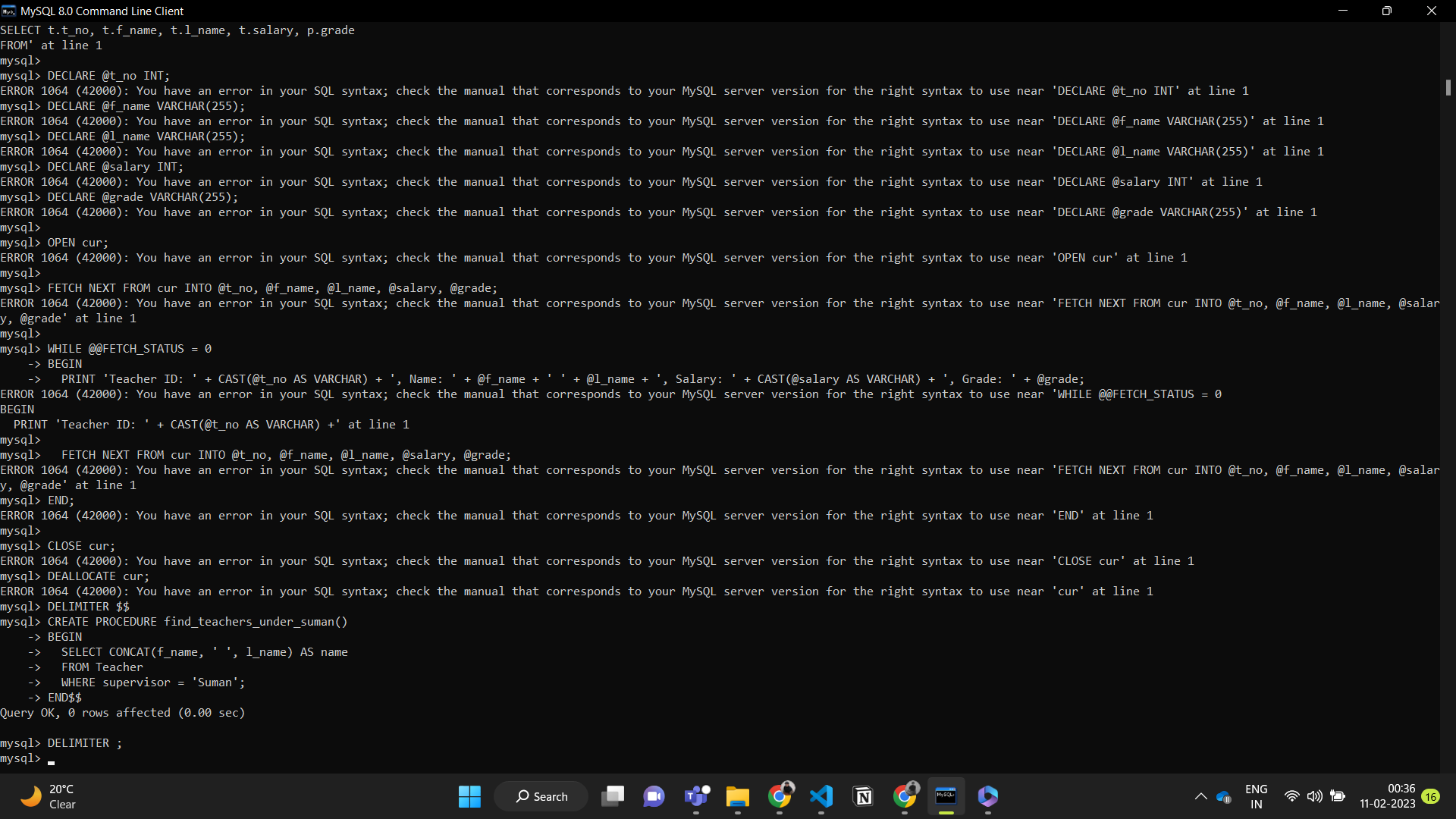
FROM Teacher

WHERE supervisor = 'Suman';

END$$

DELIMITER ;

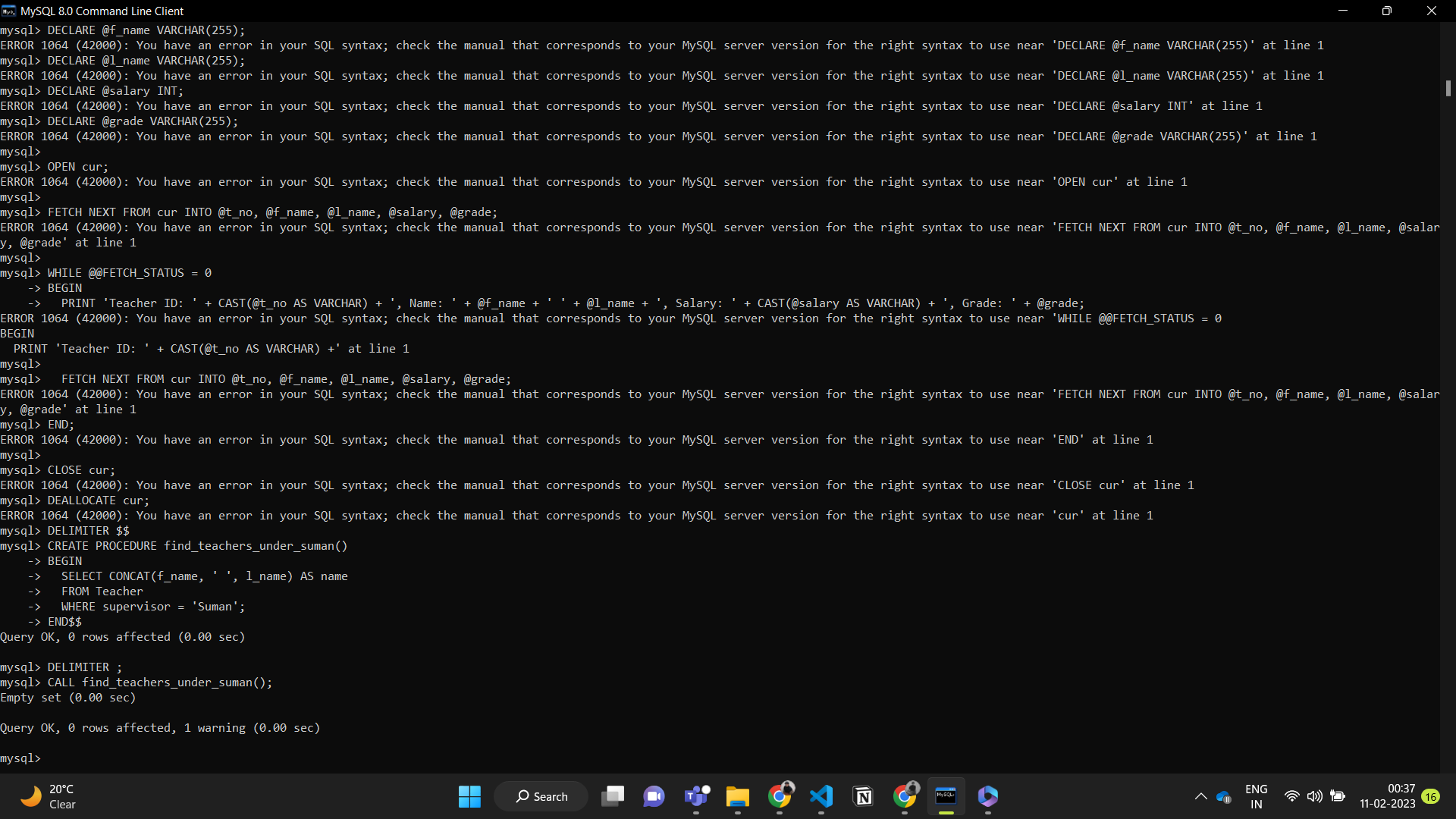
**OUTPUT:**



CODE FOR CALL THE PROCEDURE:

CALL find\_teachers\_under\_suman();

**OUTPUT:**



**PROBLEM 1:**

**3. Create programs for following for above given schemas**

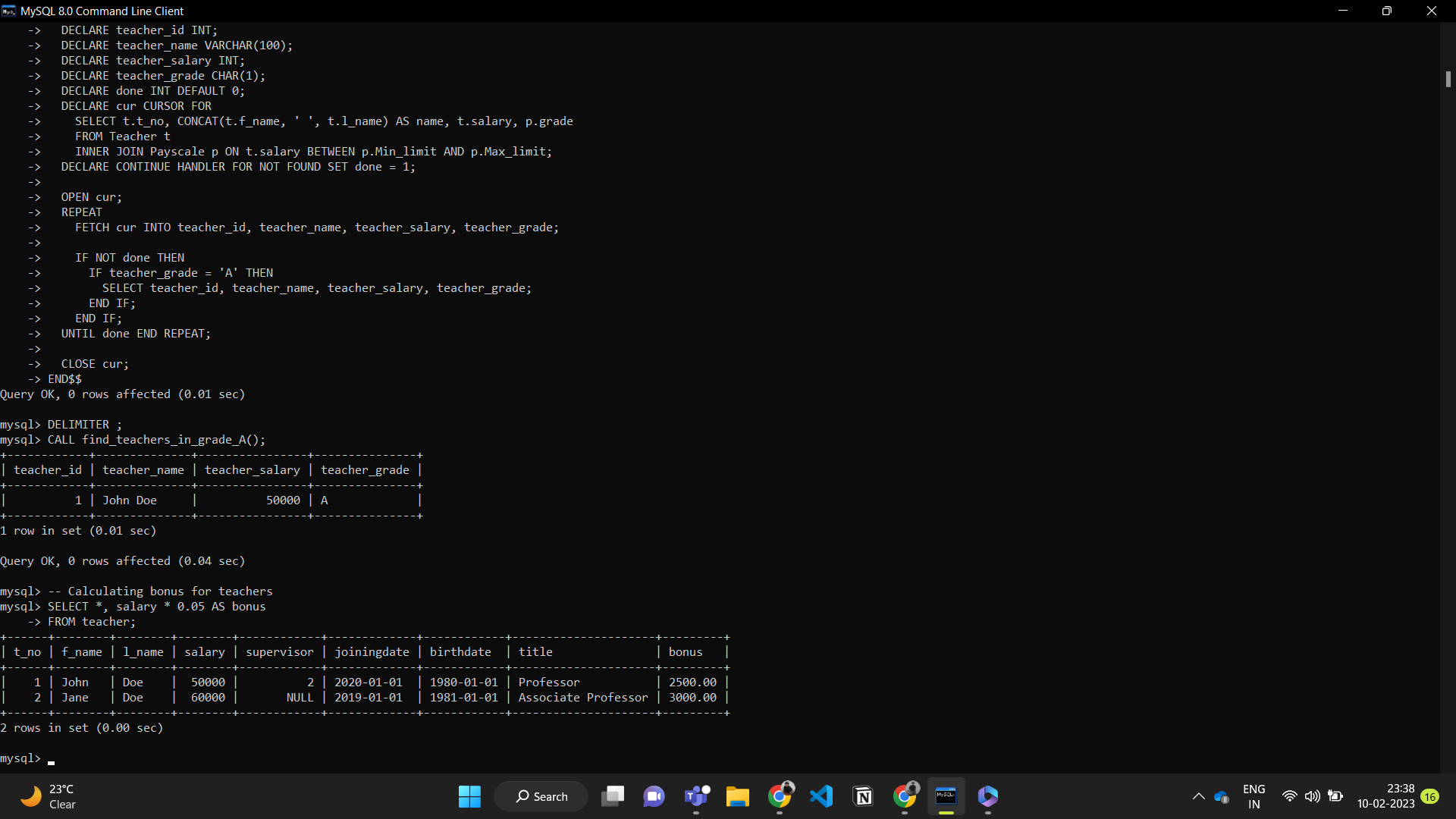
**a. Create a host language block using a cursor to calculate bonus for teachers as 5% of their salary. Display on screen the teacher details along with the bonus given.**

**CODE:**

SELECT \*, salary \* 0.05 AS bonus

FROM teacher;

**OUTPUT:**



**b. Write a host language block to delete all the rows from the ‘teacher’ table where the salary is less than Rs.5000.**

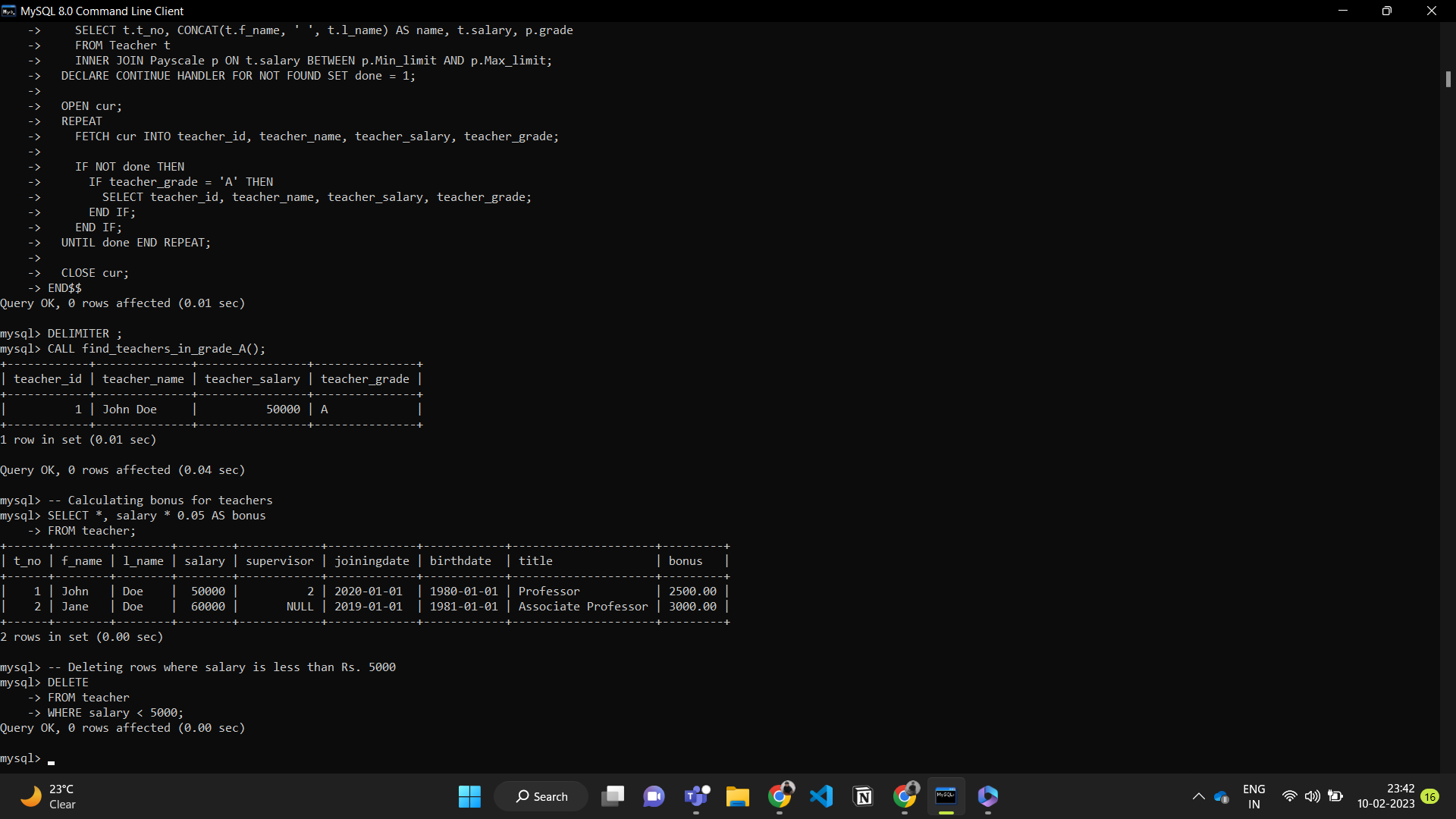
**CODE:**

DELETE

FROM teacher

WHERE salary < 5000;

**OUTPUT:**



**c. Write a host language code to insert the supervisor information from ‘teacher’ table to another table called ‘supervisor’. The new table should have only those records where the job title is ‘supervisor’.**

**CODE:**

CREATE TABLE supervisor (

t\_no INT PRIMARY KEY,

f\_name VARCHAR(255),

l\_name VARCHAR(255),

supervisor VARCHAR(255)

);

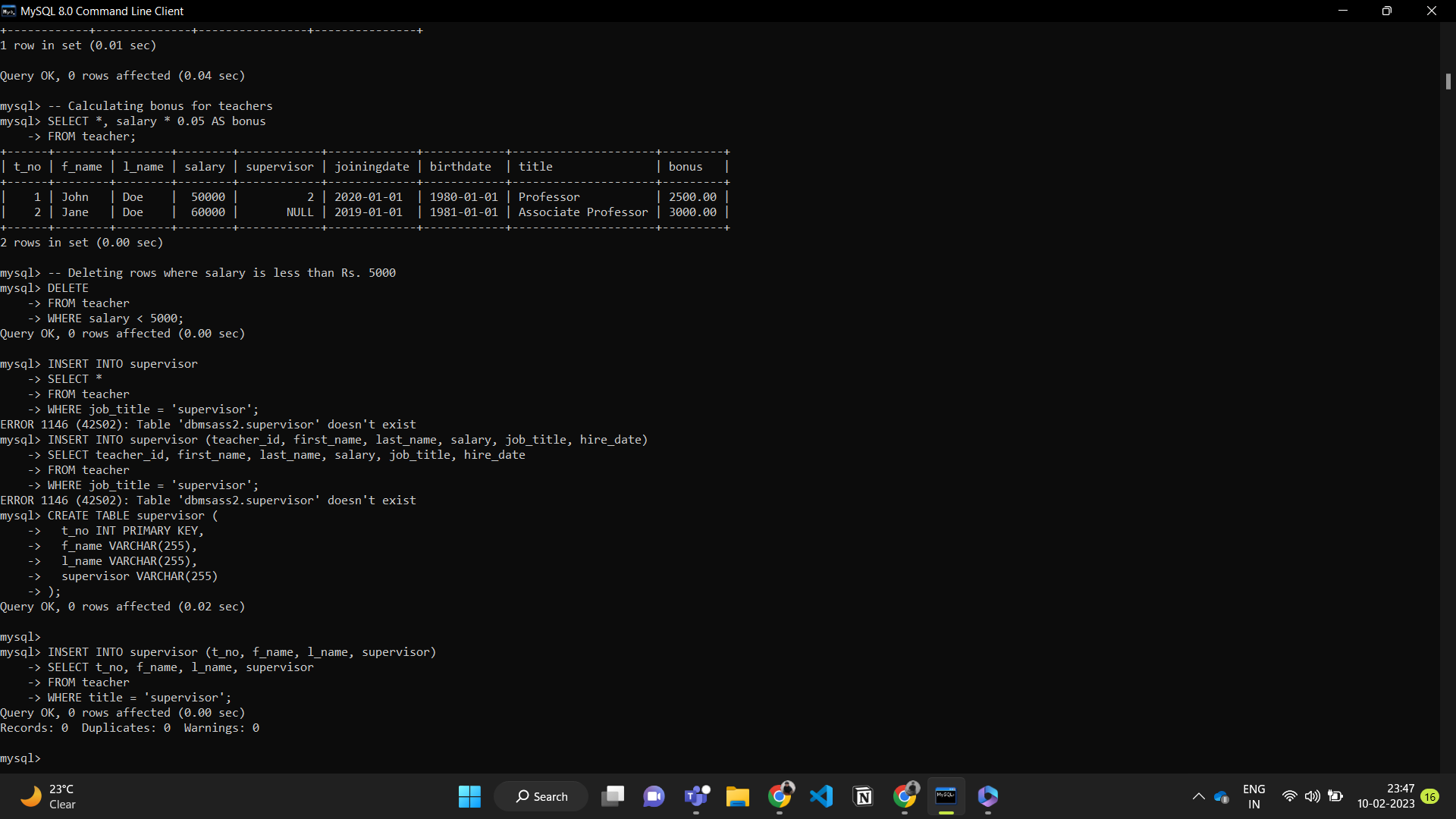
INSERT INTO supervisor (t\_no, f\_name, l\_name, supervisor)

SELECT t\_no, f\_name, l\_name, supervisor

FROM teacher

WHERE title = 'supervisor';

**OUTPUT:**



**d. Write a block in host language that deletes all the rows from ‘teacher’ table if the teacher was hired for more than 10 years.**

**CODE:**

DELETE FROM teacher

WHERE (YEAR(CURDATE()) - YEAR(joiningdate)) > 10;

**OUTPUT:**

