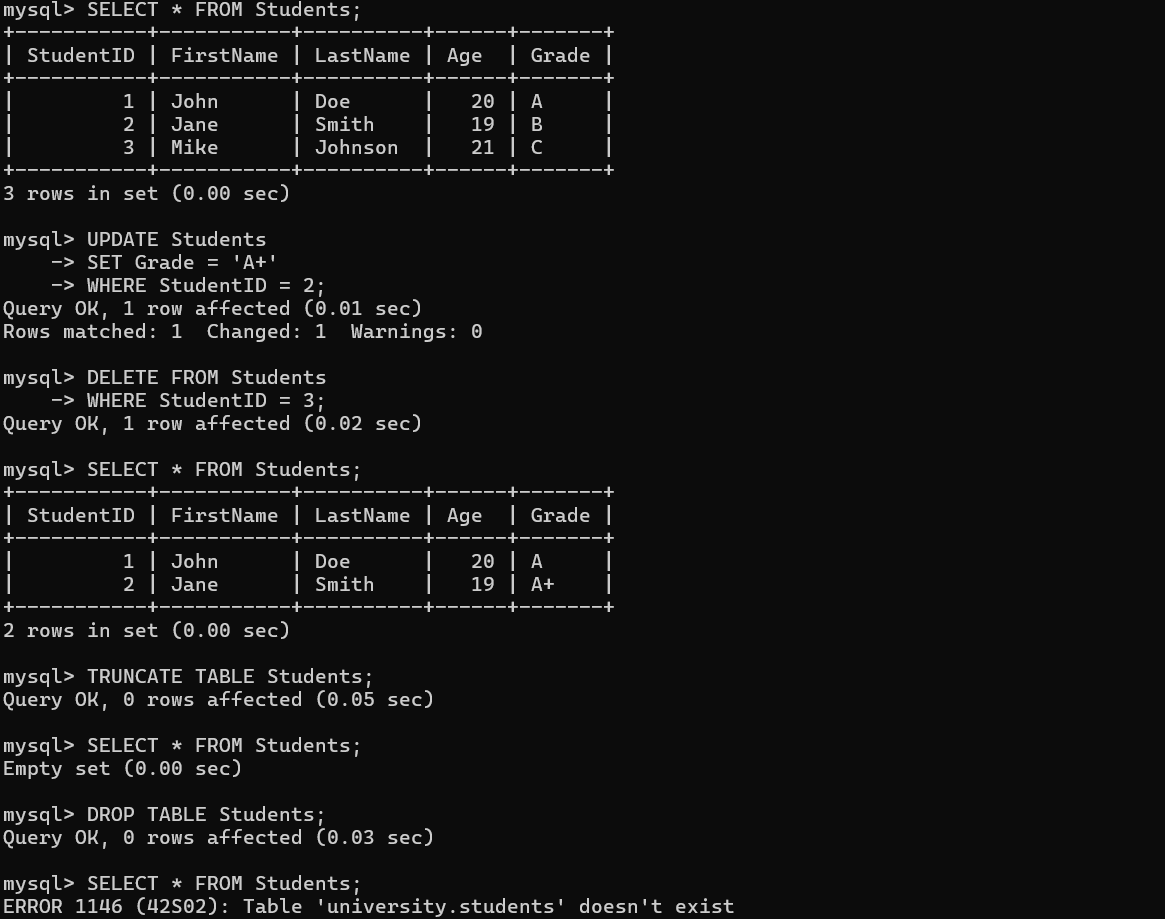
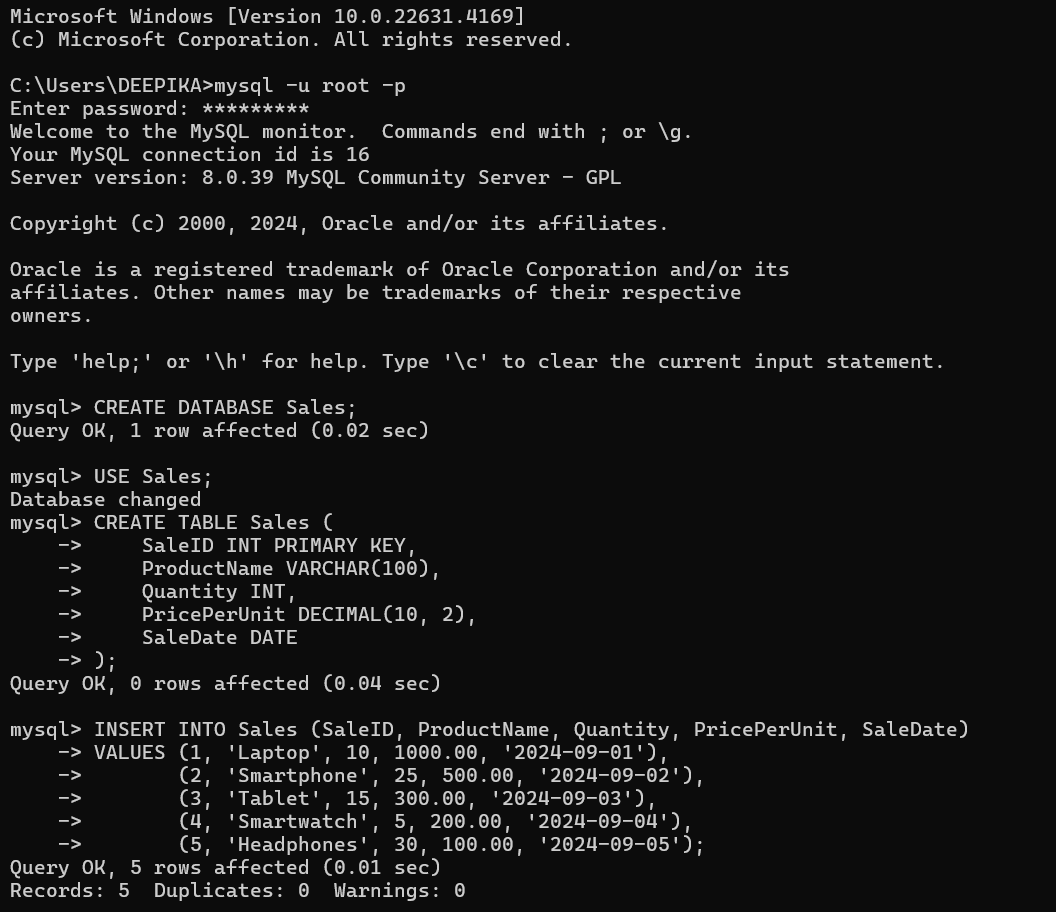
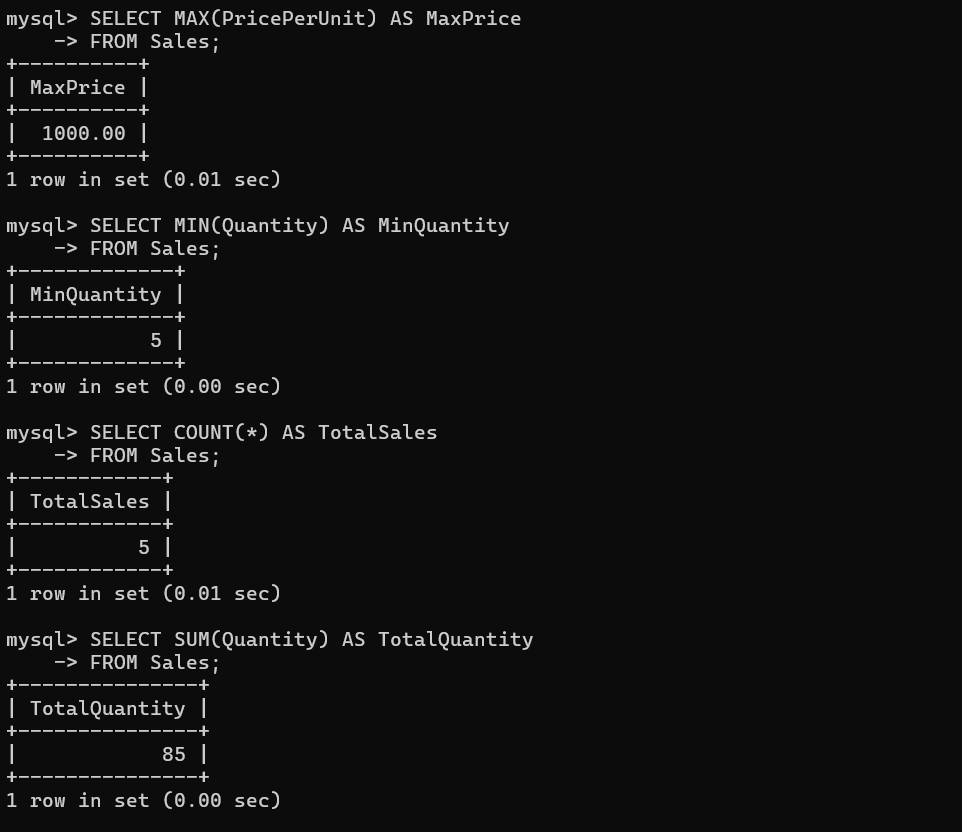
**SOURCE CODE & OUTPUT:**

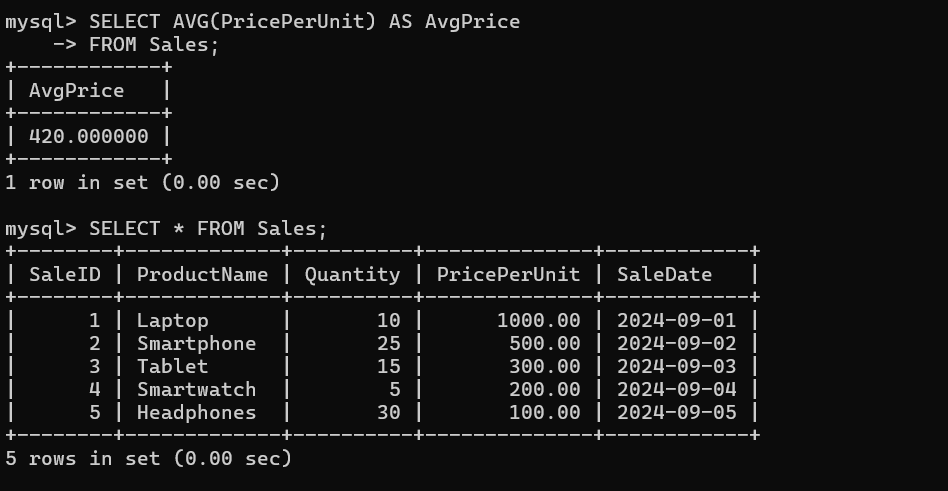
****

****

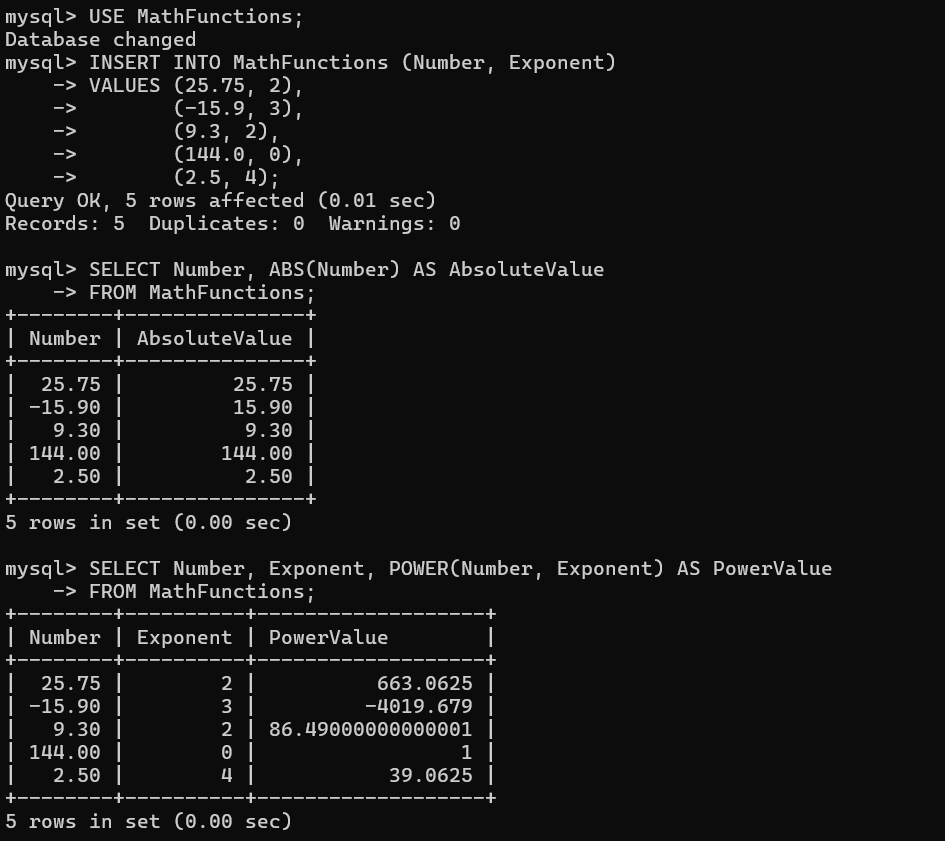
**SOURCE CODE & OUTPUT:**

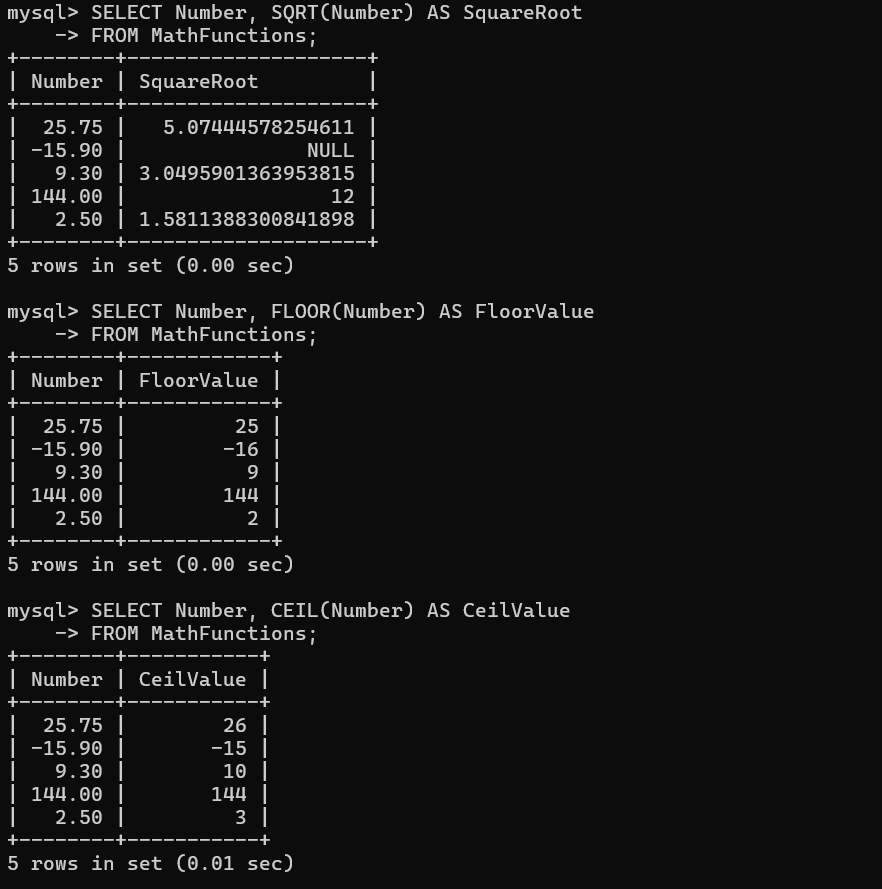
****

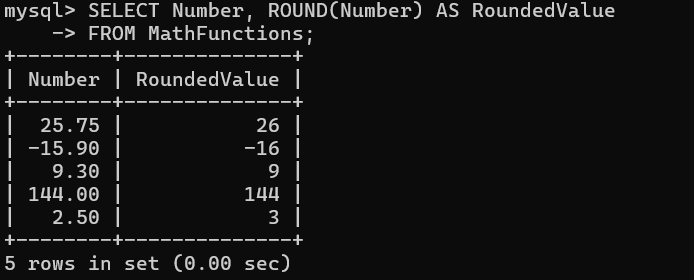
****

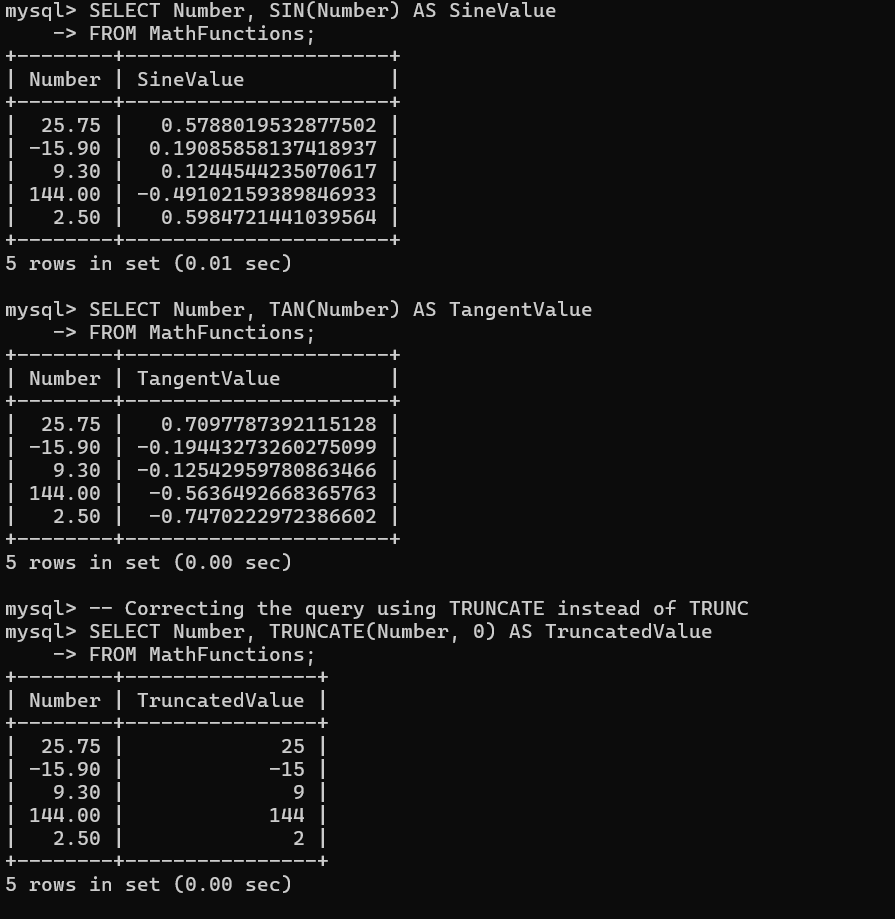
****

**SOURCE CODE & OUTPUT:**

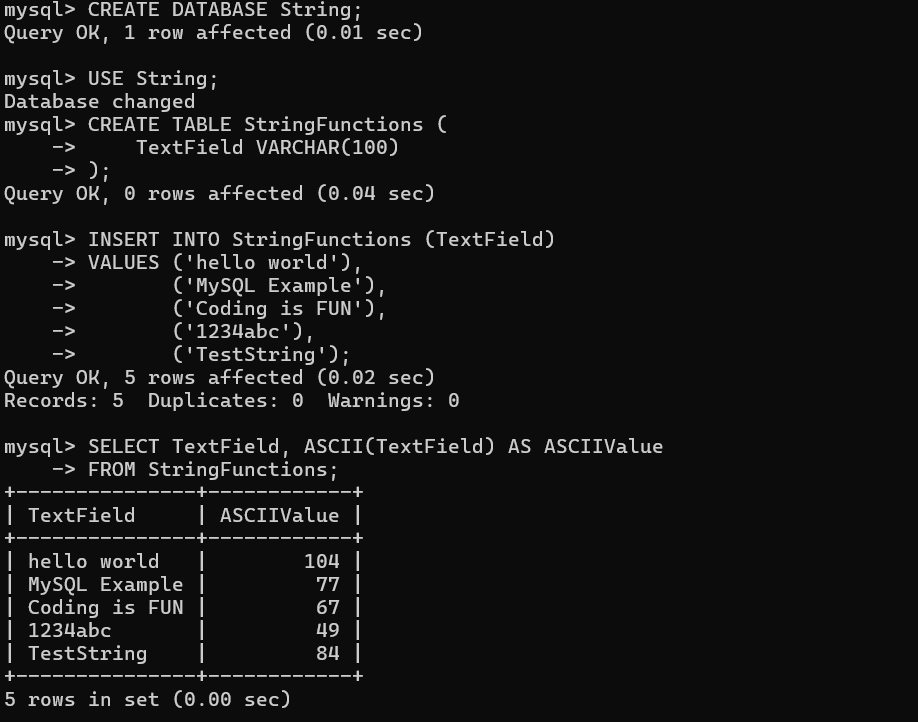
****

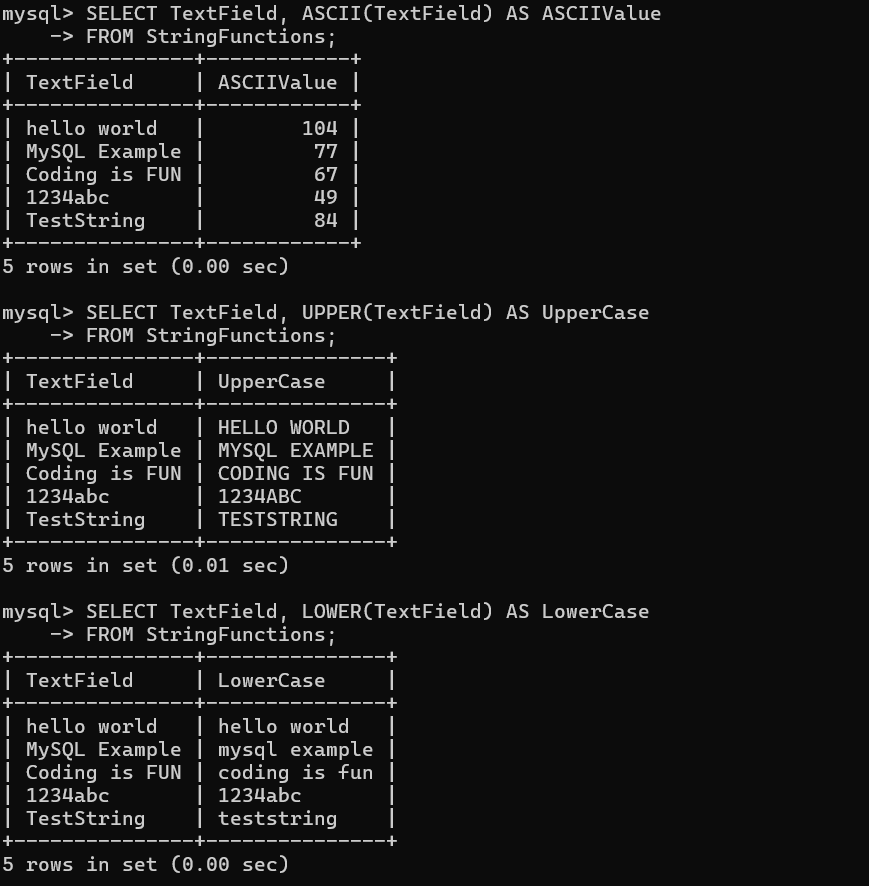
****

****

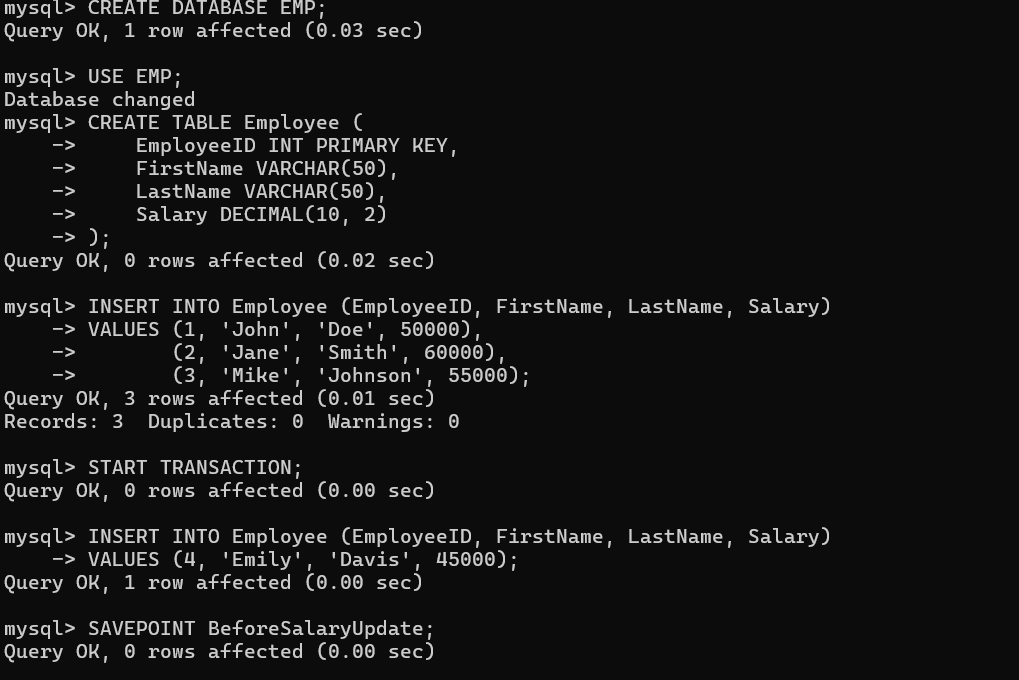
****

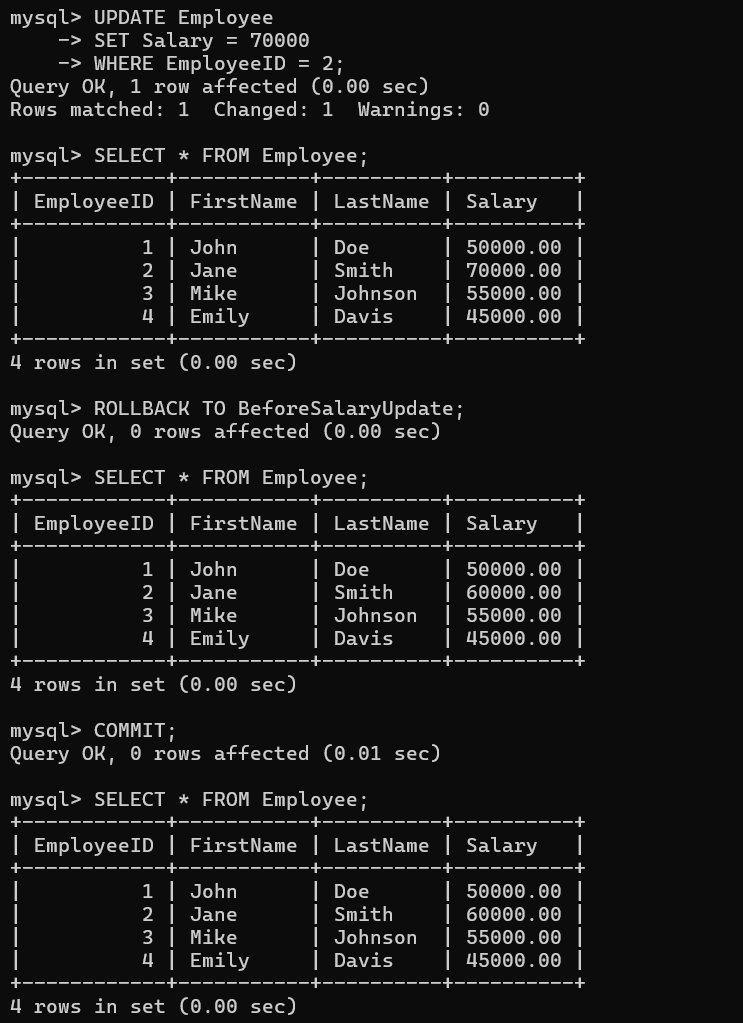
**SOURCE CODE & OUTPUT:**

****

****

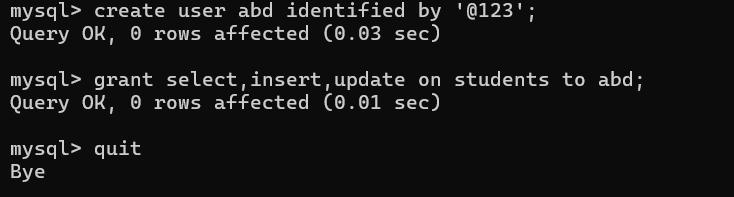
**SOURCE CODE & OUTPUT:**

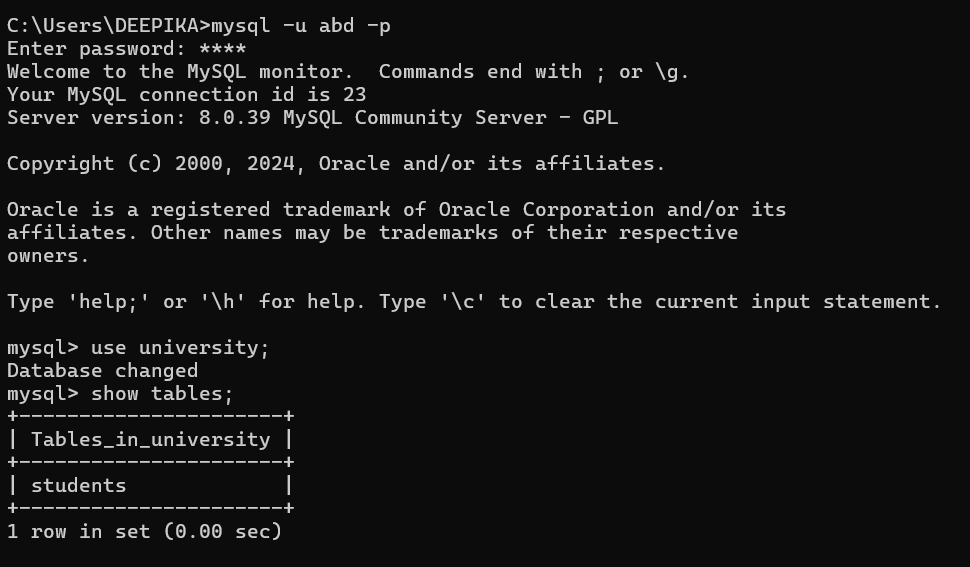
****

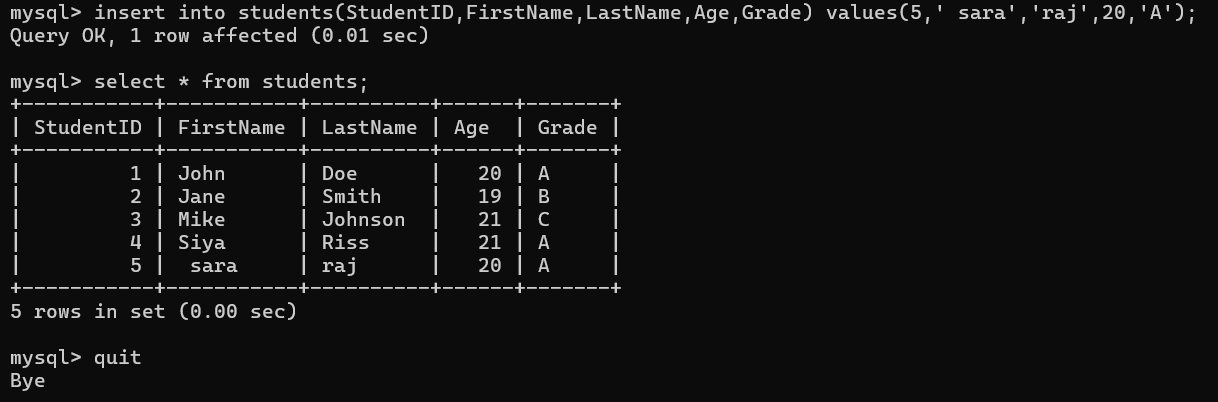
****

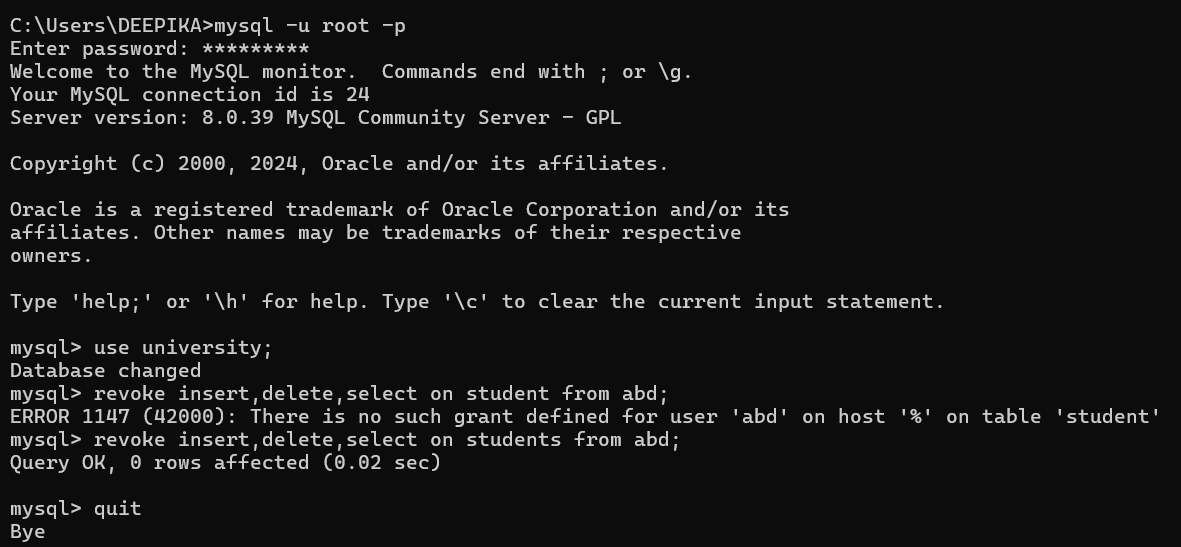
**SOURCE CODE & OUTPUT:**

****

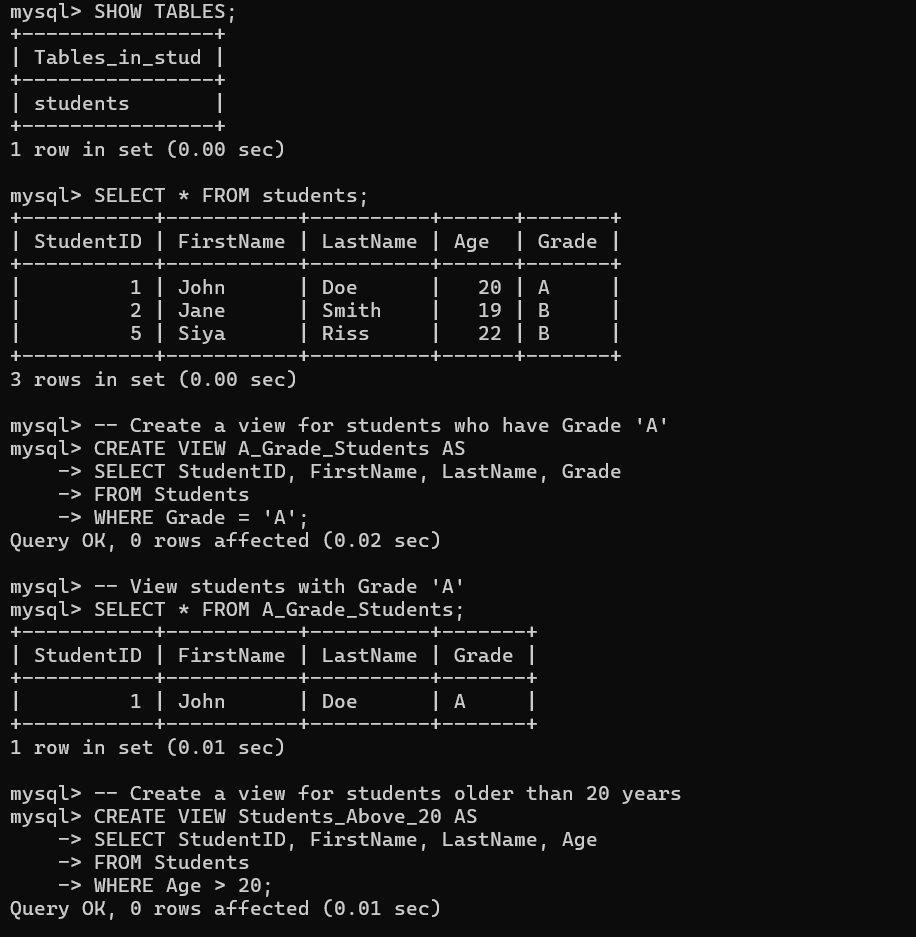
****

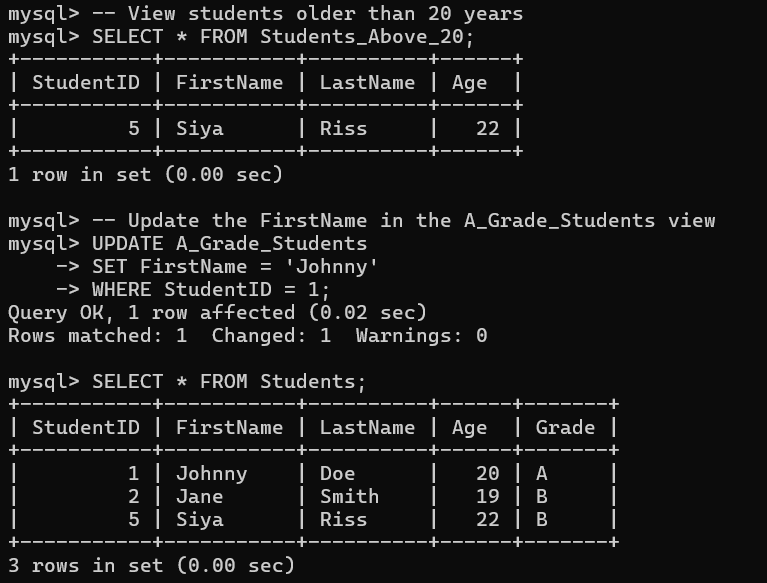
****

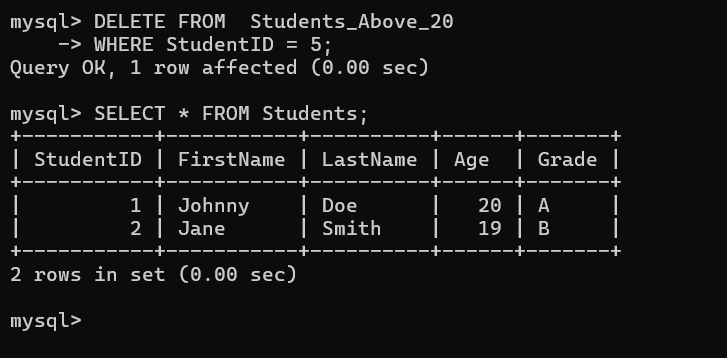
****

****

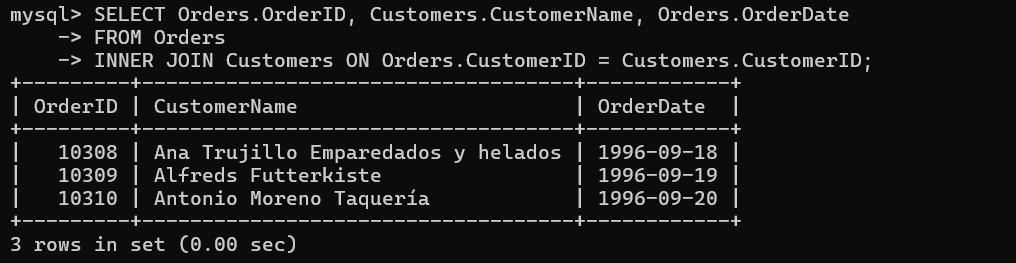
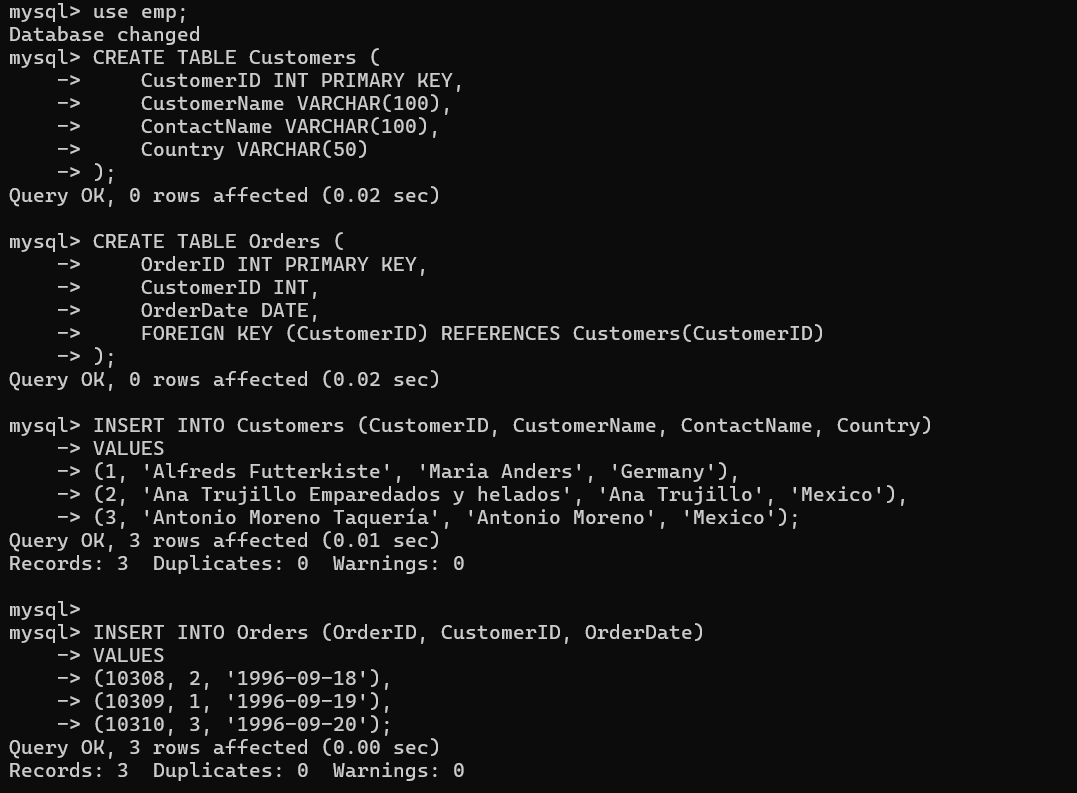
**SOURCE CODE & OUTPUT:**

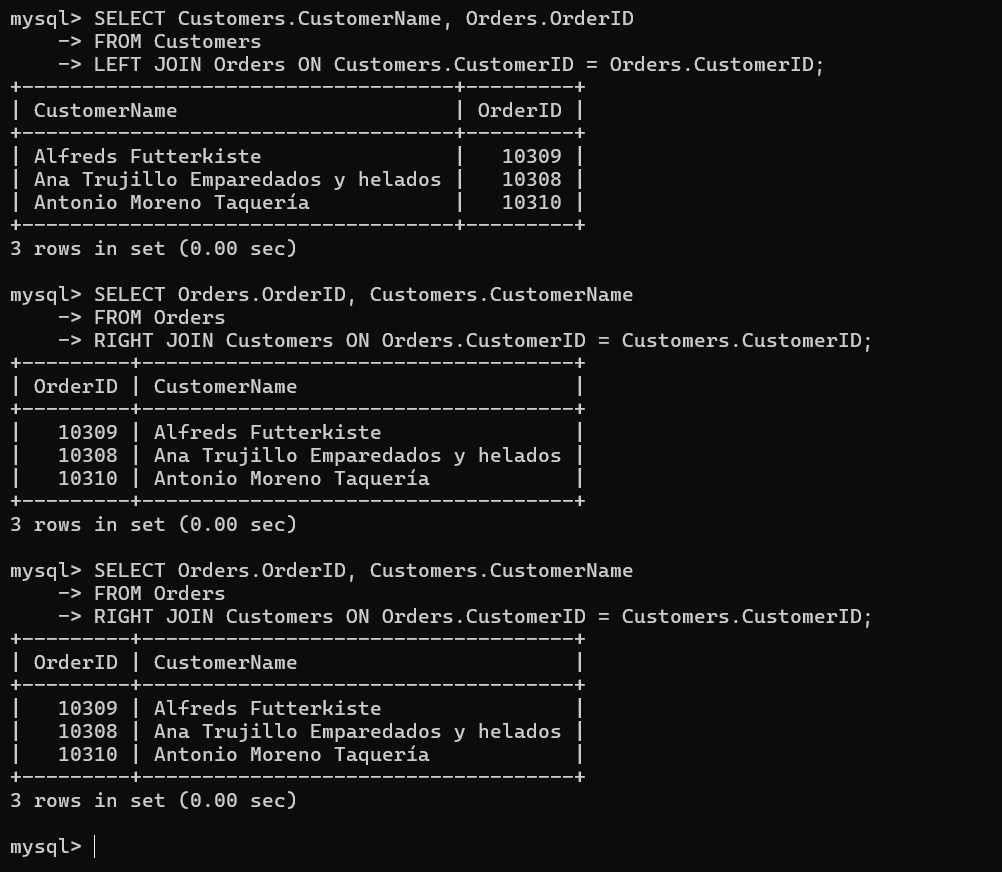
****

****

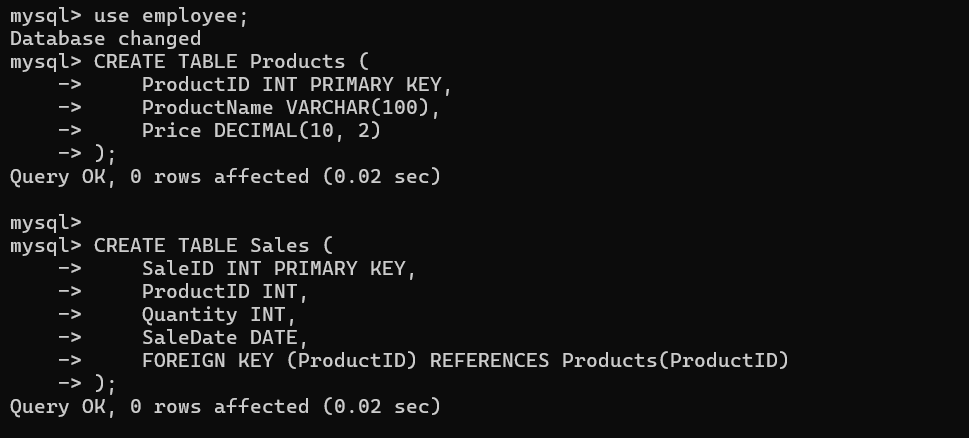
****

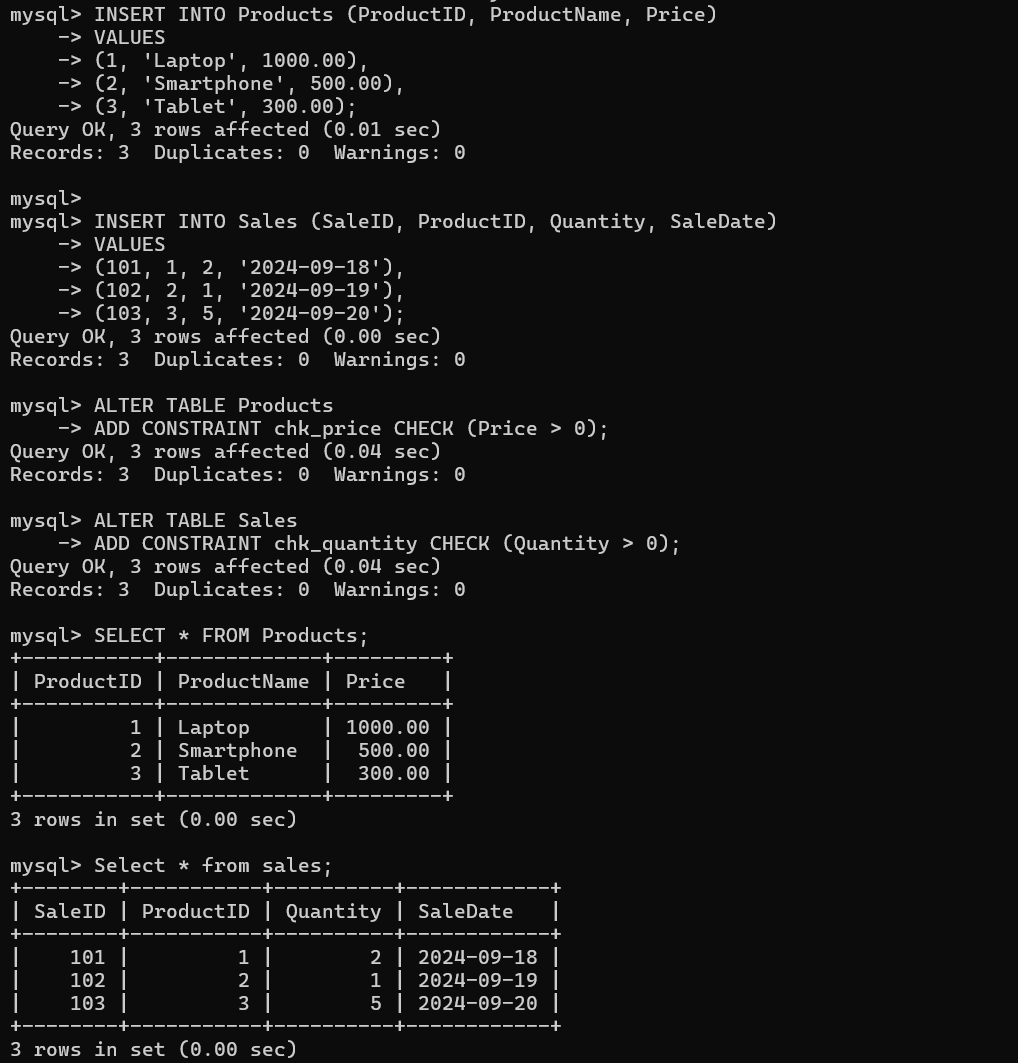
**SOURCE CODE & OUTPUT:**

****

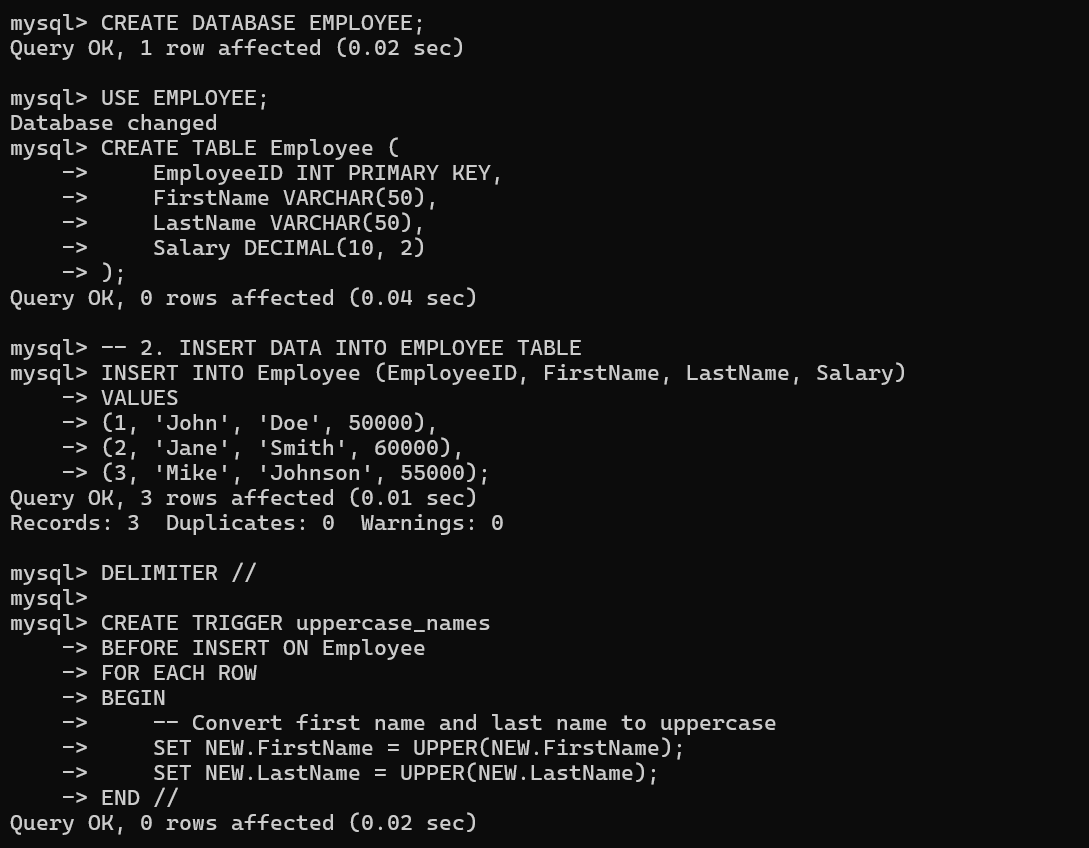
****

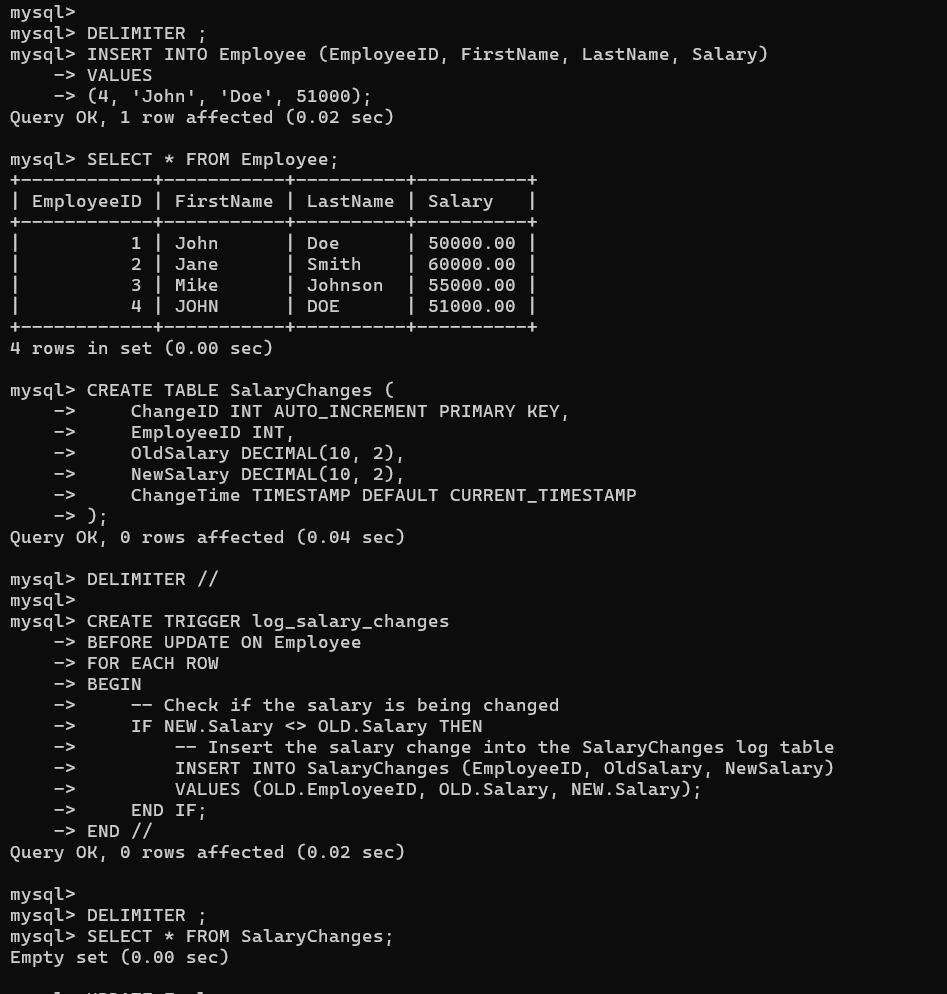
**SOURCE CODE & OUTPUT:**

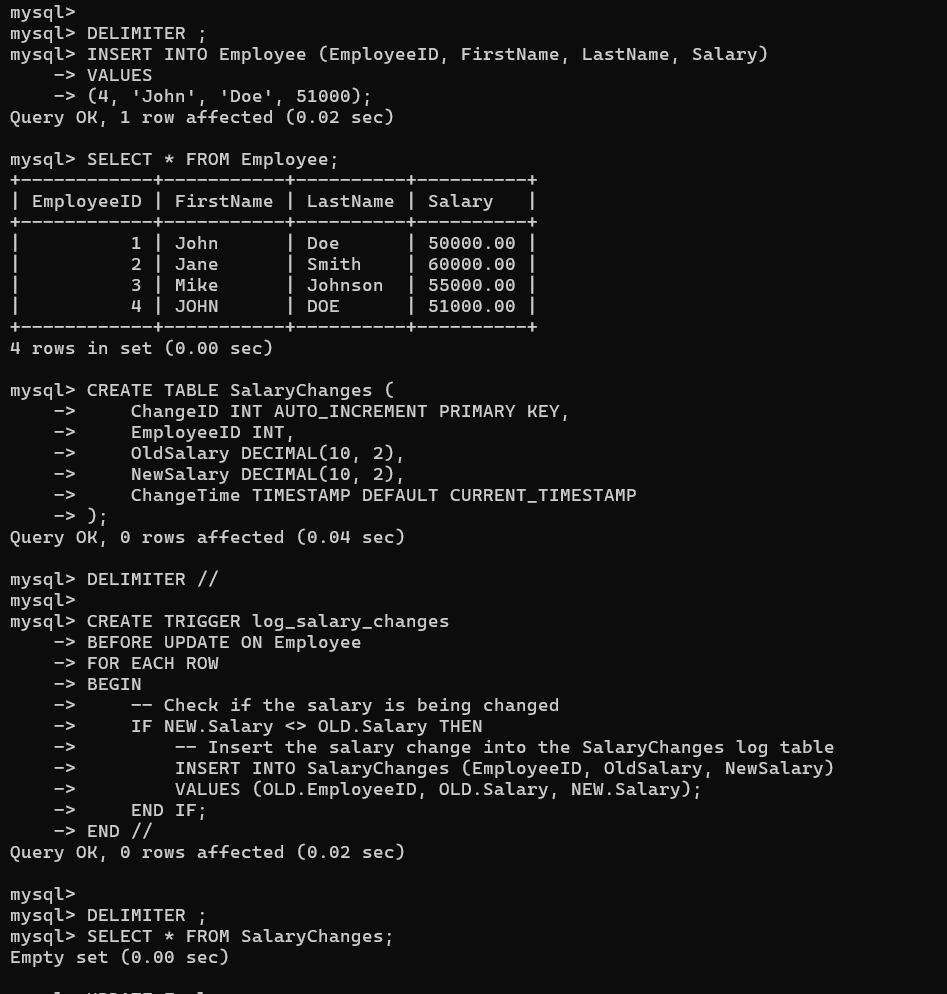
****

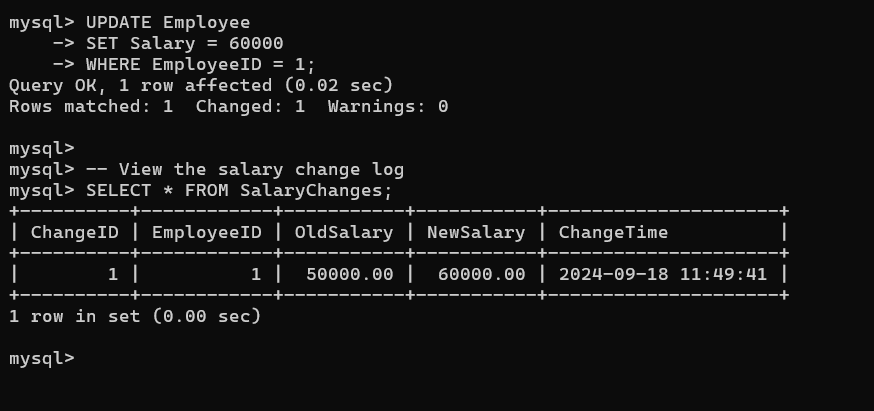
****

**SOURCE CODE & OUTPUT:**

****

****

****

****

**SOURCE CODE:**DECLARE

total\_rows NUMBER(2);

BEGIN

UPDATE customers

SET salary = salary + 500;

IF SQL%NOTFOUND THEN

DBMS\_OUTPUT.PUT\_LINE('No customers updated');

ELSE

total\_rows := SQL%ROWCOUNT;

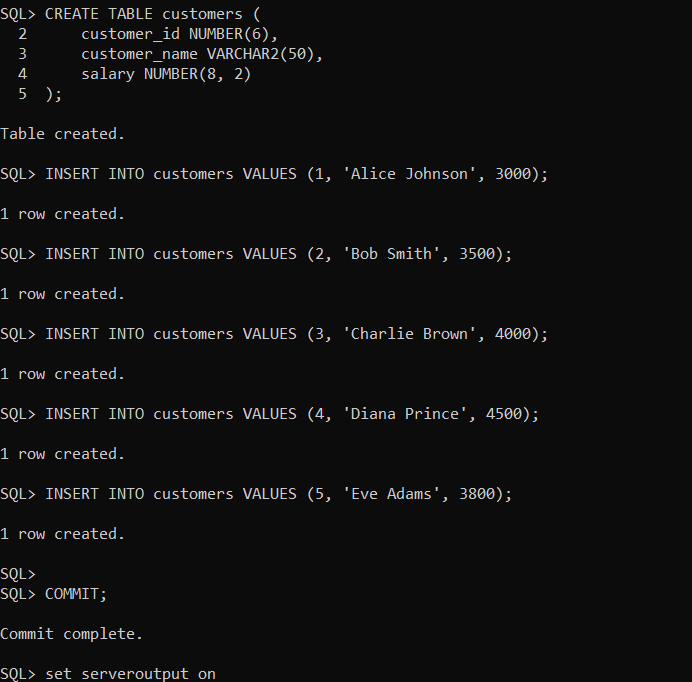
DBMS\_OUTPUT.PUT\_LINE(total\_rows || ' customers updated.');

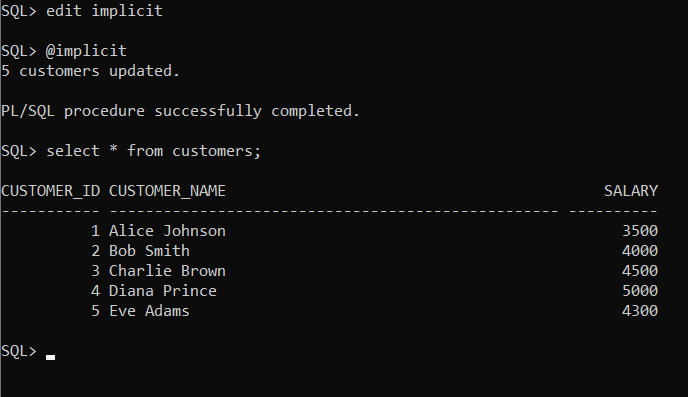
END IF;

END;

/

**OUTPUT:**

****

****

**SOURCE CODE;**

DECLARE

CURSOR c\_customer IS

SELECT name, address FROM customer;

v\_name customer.name%TYPE;

v\_address customer.address%TYPE;

BEGIN

OPEN c\_customer;

LOOP

FETCH c\_customer INTO v\_name, v\_address;

EXIT WHEN c\_customer%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE('Name: ' || v\_name);

DBMS\_OUTPUT.PUT\_LINE('Address: ' || v\_address);

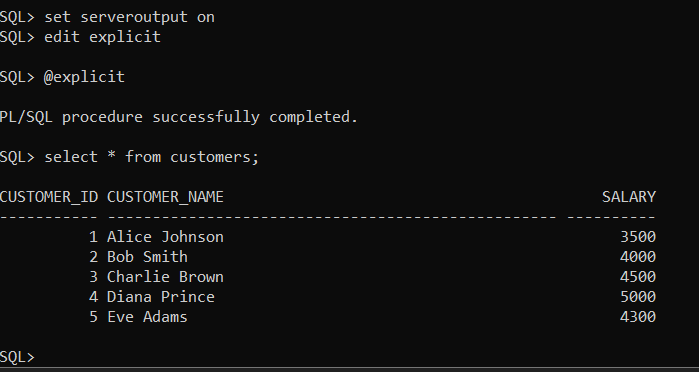
END LOOP;

CLOSE c\_customer;

END;

/

**OUTPUT:**

****

**SOURCE CODE:**

DECLARE

v\_customer\_name CUSTOMERS.CUSTOMER\_NAME%TYPE;

BEGIN

SELECT CUSTOMER\_NAME

INTO v\_customer\_name

FROM CUSTOMERS

WHERE CUSTOMER\_ID = 101; -- Assumed ID for demonstration

DBMS\_OUTPUT.PUT\_LINE('Customer Name: ' || v\_customer\_name);

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('Error: No customer found with the given ID.');

WHEN TOO\_MANY\_ROWS THEN

DBMS\_OUTPUT.PUT\_LINE('Error: More than one customer found with the given ID.');

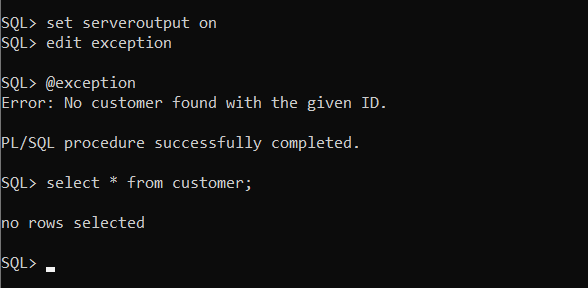
WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('An unexpected error occurred: ' || SQLERRM);

END;

/

**OUTPUT:**

****

**SOURCE CODE:**

DECLARE

c\_id clients.id%TYPE := &cc\_id;

c\_name clients.name%TYPE;

c\_addr clients.address%TYPE;

ex\_invalid\_id EXCEPTION;

BEGIN

IF c\_id <= 0 THEN

RAISE ex\_invalid\_id;

ELSE

SELECT name, address

INTO c\_name, c\_addr

FROM clients

WHERE id = c\_id;

DBMS\_OUTPUT.PUT\_LINE('Name: ' || c\_name);

DBMS\_OUTPUT.PUT\_LINE('Address: ' || c\_addr);

END IF;

EXCEPTION

WHEN ex\_invalid\_id THEN

DBMS\_OUTPUT.PUT\_LINE('ID must be greater than zero!');

ID

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('No such client!');

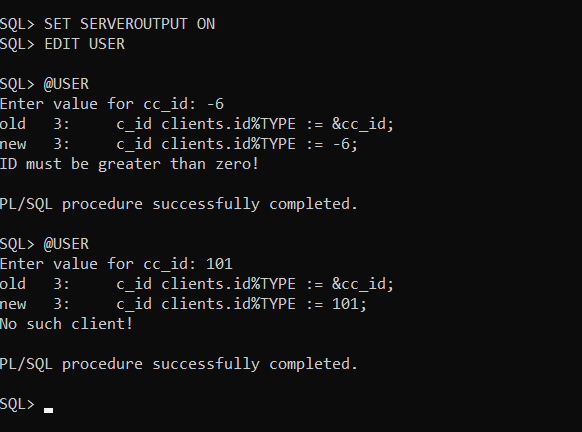
WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('Error: ' || SQLERRM);

END;

/

**OUTPUT:**

****

**SOURCE CODE:**

CREATE TABLE sum\_results (

id NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,

num1 NUMBER,

num2 NUMBER,

result NUMBER

);

CREATE OR REPLACE FUNCTION calculate\_sum(a IN NUMBER, b IN NUMBER) RETURN NUMBER IS

c NUMBER;

BEGIN

c := a + b;

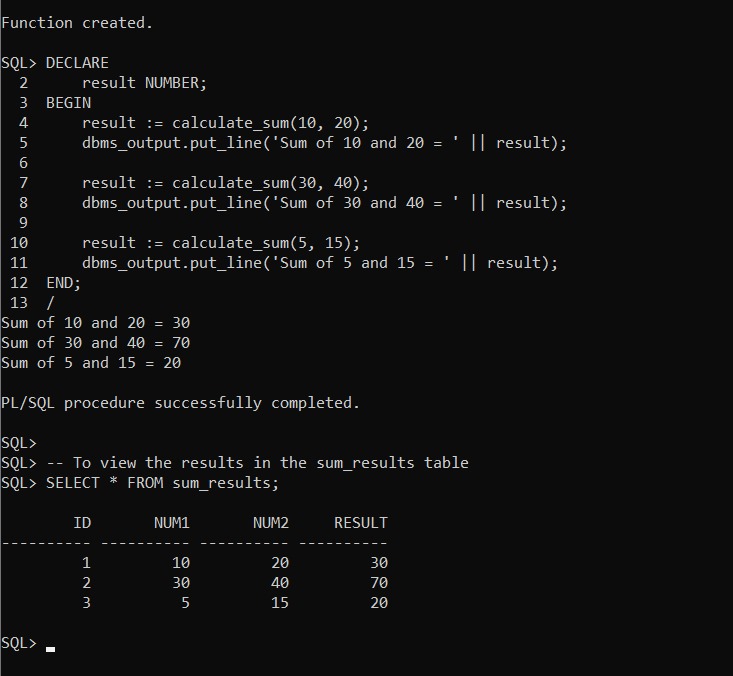
INSERT INTO sum\_results (num1, num2, result) VALUES (a, b, c);

RETURN c;

END calculate\_sum;

/

**OUTPUT:**



**SOURCE CODE:**

CREATE OR REPLACE PROCEDURE calculate\_sum\_proc(a IN NUMBER, b IN NUMBER) IS

c NUMBER;

BEGIN

c := a + b;

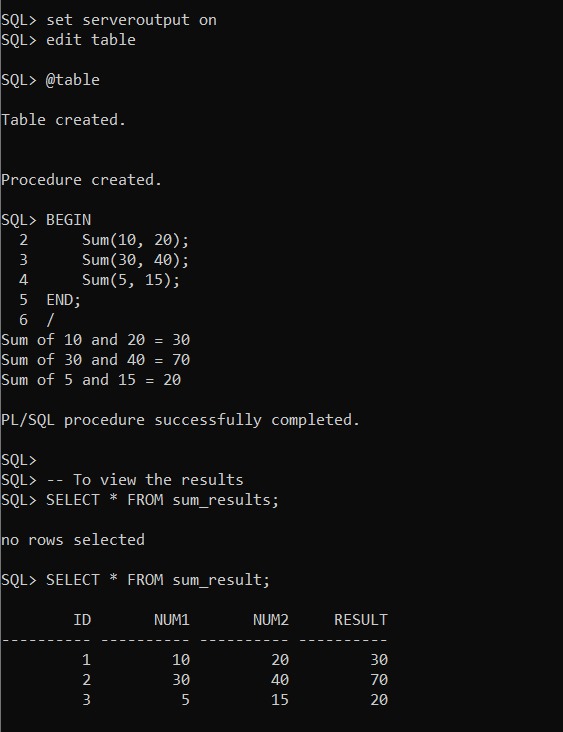
INSERT INTO sum\_results (num1, num2, result) VALUES (a, b, c);

DBMS\_OUTPUT.PUT\_LINE('The sum of ' || a || ' and ' || b || ' is: ' || c);

END calculate\_sum\_proc;

/

**OUTPUT:**

****