**MySQL**

**Exercise 1**

**1. Write a program that computes the perimeter and the area of a rectangle. Define your own values for the length and width. (Assuming that L and W are the length and width of the rectangle, Perimeter = 2\*(L+W) and Area = L\*W.**

mysql> create table rectangle(

-> length double,

-> width double,

-> perimiter double,

-> area double

-> );

Query OK, 0 rows affected (0.20 sec)

mysql> delimiter //

mysql> create procedure rectangle()

-> begin

-> declare l double default 10.0;

-> declare w double default 5.0;

-> declare perimeter double;

-> declare area double;

-> set perimeter = 2\*(l+w);

-> set area = l\*w;

-> insert into rectangle(length,width,perimiter,area)

-> values(l,w,perimeter,area);

-> end//

Query OK, 0 rows affected (0.03 sec)

mysql> delimiter ;

mysql> call rectangle();

Query OK, 1 row affected (0.04 sec)

mysql> select \* from rectangle;

+--------+-------+-----------+------+

| length | width | perimiter | area |

+--------+-------+-----------+------+

| 10 | 5 | 30 | 50 |

+--------+-------+-----------+------+

1 row in set (0.01 sec)

****

**2. Write a program that declares an integer variable called num, assigns a value to it, and computes and inserts into the tempp table the value of the variable itself, its square, and its cube.**

mysql> create table temp(

-> num int,

-> square int,

-> `cube` int

-> );

Query OK, 0 rows affected (0.07 sec)

mysql> delimiter //

mysql> create procedure squareandcube(in num int)

-> begin

-> declare square int;

-> declare `cube` int;

-> set square = num \* num;

-> set `cube` = num \* num \* num;

-> insert into temp values(num,square,`cube`);

-> end;//

Query OK, 0 rows affected (0.01 sec)

mysql> delimiter ;

mysql> call squareandcube(5);

Query OK, 1 row affected (0.02 sec)

mysql> select \* from temp;

+------+--------+------+

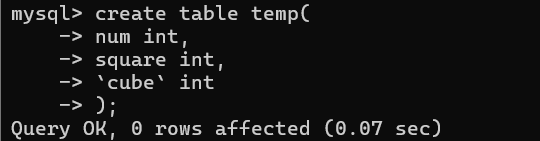
| num | square | cube |

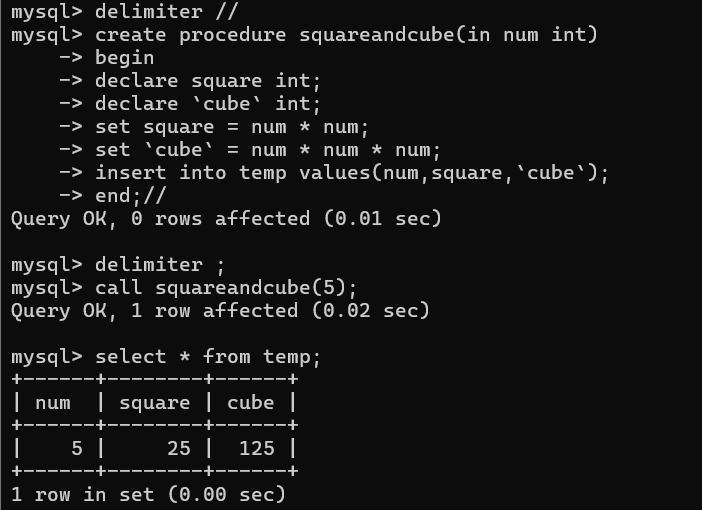
+------+--------+------+

| 5 | 25 | 125 |

+------+--------+------+

1 row in set (0.00 sec)





**3. Convert a temperature in Fahrenheit (F) to its equivalent in Celsius (C) and vice versa. The required formulae are:- C= (F-32)\*5/9 F= 9/5\*C + 32**

mysql> DELIMITER //

mysql>

mysql> CREATE PROCEDURE Temperature(IN input\_temp DOUBLE, IN unit CHAR(1))

-> BEGIN

-> DECLARE converted\_temp DOUBLE;

->

-> IF unit = 'C' THEN

-> -- Convert from Celsius to Fahrenheit

-> SET converted\_temp = (input\_temp \* 9/5) + 32;

-> SELECT CONCAT(input\_temp, ' °C is equivalent to ', converted\_temp, ' °F') AS ConversionResult;

->

-> ELSEIF unit = 'F' THEN

-> -- Convert from Fahrenheit to Celsius

-> SET converted\_temp = (input\_temp - 32) \* 5/9;

-> SELECT CONCAT(input\_temp, ' °F is equivalent to ', converted\_temp, ' °C') AS ConversionResult;

->

-> ELSE

-> -- If invalid unit provided

-> SELECT 'Invalid unit. Use "C" for Celsius or "F" for Fahrenheit.' AS ErrorMessage;

-> END IF;

-> END //

Query OK, 0 rows affected (0.02 sec)

mysql>

mysql> DELIMITER ;

mysql>

mysql>

mysql> CALL Temperature(25, 'C');

+------------------------------+

| ConversionResult |

+------------------------------+

| 25 °C is equivalent to 77 °F |

+------------------------------+

1 row in set (0.01 sec)

Query OK, 0 rows affected (0.01 sec)

mysql> CALL Temperature(77, 'F');

+------------------------------+

| ConversionResult |

+------------------------------+

| 77 °F is equivalent to 25 °C |

+------------------------------+

1 row in set (0.00 sec)

Query OK, 0 rows affected (0.01 sec)

mysql> CALL Temperature(100, 'X');

+----------------------------------------------------------+

| ErrorMessage |

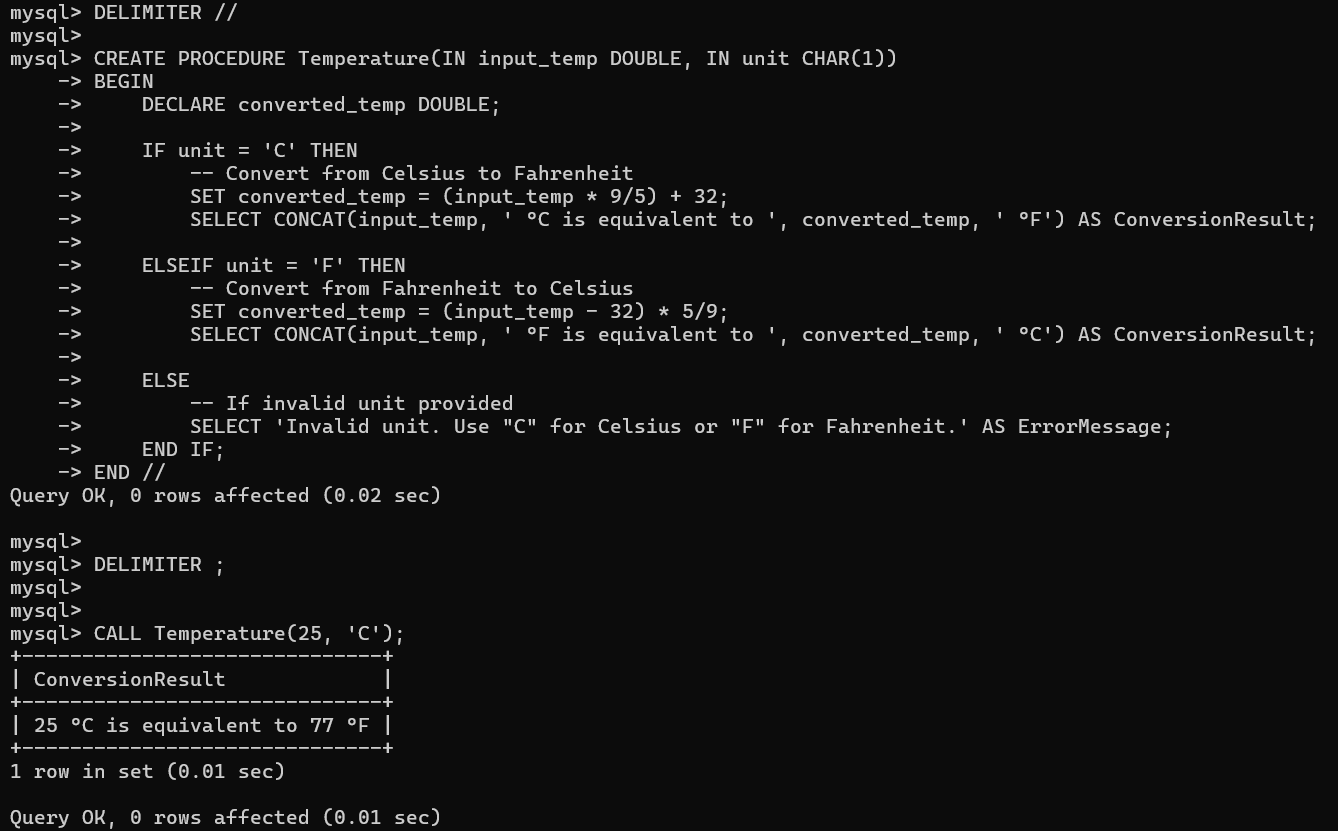
+----------------------------------------------------------+

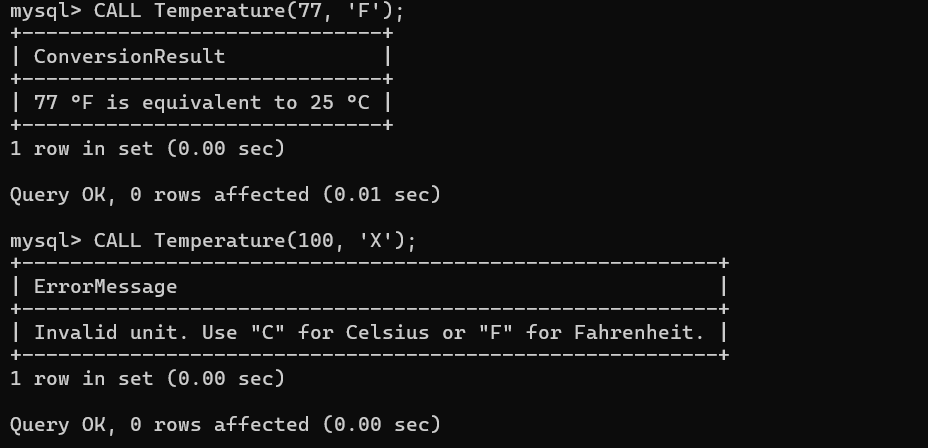
| Invalid unit. Use "C" for Celsius or "F" for Fahrenheit. |

+----------------------------------------------------------+

1 row in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

****

****

**4. Convert a number of inches into yards, feet, and inches. For example, 124 inches equals 3 yards, 1 foot, and 4 inches.**

mysql> create table conversion(

-> total\_inches INT,

-> yards INT,

-> feet INT,

-> inches INT

-> );

Query OK, 0 rows affected (0.05 sec)

mysql> delimiter //

mysql> create procedure InchesToYardFeetInch(IN total\_inches INT)

-> begin

-> declare yards int;

-> declare feet int;

-> declare inches int;

-> declare remaining\_inches int;

-> set yards = total\_inches DIV 36;

-> set remaining\_inches = total\_inches MOD 36;

-> set feet = remaining\_inches DIV 12;

-> set inches = remaining\_inches MOD 12;

-> insert into conversion values(total\_inches, yards, feet, inches);

-> end;//

Query OK, 0 rows affected (0.01 sec)

mysql> delimiter ;

mysql> call InchesToYardFeetInch(124);

Query OK, 1 row affected (0.01 sec)

mysql> select \* from conversion;

+--------------+-------+------+--------+

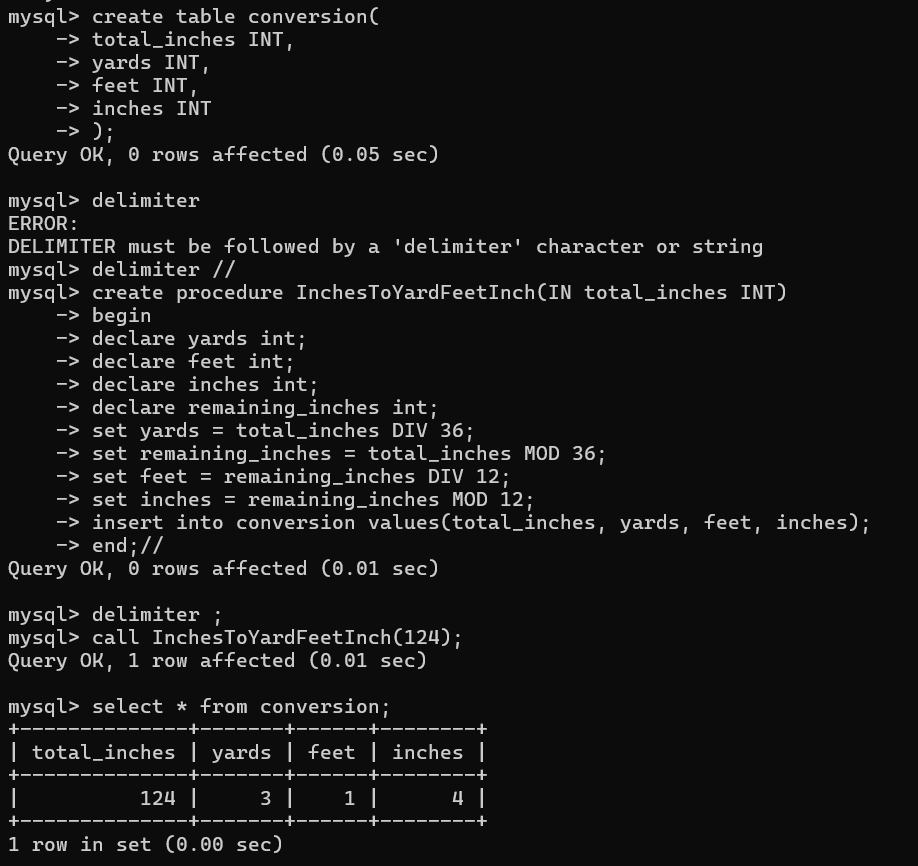
| total\_inches | yards | feet | inches |

+--------------+-------+------+--------+

| 124 | 3 | 1 | 4 |

+--------------+-------+------+--------+

1 row in set (0.00 sec)

****

**5. Write a program that enables a user to input an integer. The program should then state whether the integer is evenly divisible by 5.**

mysql> DELIMITER //

mysql> CREATE PROCEDURE DivisibleBy5(IN num INT)

-> BEGIN

-> IF num MOD 5 = 0 THEN

-> SELECT CONCAT(num, ' is divisible by 5.') AS Result;

-> ELSE

-> SELECT CONCAT(num, ' is NOT divisible by 5.') AS Result;

-> END IF;

-> END;//

Query OK, 0 rows affected (0.01 sec)

mysql> DELIMITER ;

mysql> call DivisibleBy5(25);

+-----------------------+

| Result |

+-----------------------+

| 25 is divisible by 5. |

+-----------------------+

1 row in set (0.00 sec)

Query OK, 0 rows affected (0.01 sec)

mysql> call DivisibleBy5(17);

+---------------------------+

| Result |

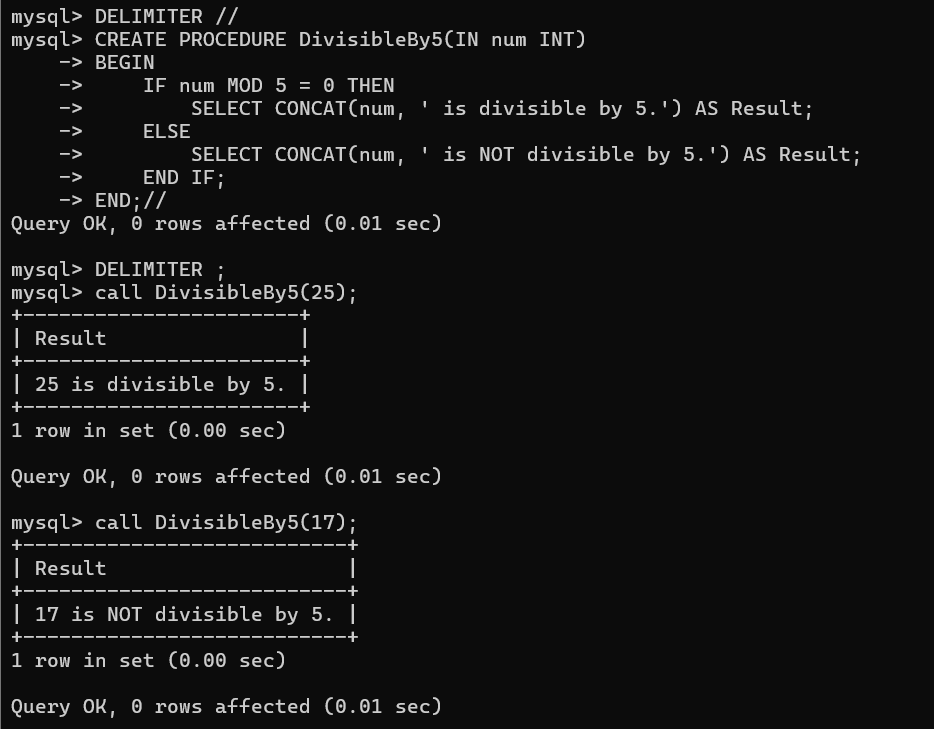
+---------------------------+

| 17 is NOT divisible by 5. |

+---------------------------+

1 row in set (0.00 sec)

Query OK, 0 rows affected (0.01 sec)

****

**6. Your block should read in two real numbers and tell whether the product of the two numbers is equal to or greater than 100.**

mysql> CREATE TABLE product(

-> num1 DOUBLE,

-> num2 DOUBLE,

-> product DOUBLE,

-> message VARCHAR(100)

-> );

Query OK, 0 rows affected (0.05 sec)

mysql> DELIMITER //

mysql> CREATE PROCEDURE CheckProduct(IN num1 DOUBLE, IN num2 DOUBLE)

-> BEGIN

-> DECLARE product DOUBLE;

-> DECLARE result BOOLEAN;

-> DECLARE msg VARCHAR(100);

-> -- Calculate the product

-> SET product = num1 \* num2;

-> -- Check the condition and set message

-> IF product >= 100 THEN

-> SET result = TRUE;

-> SET msg = CONCAT('Product is ', product, ' which is equal to or greater than 100.');

-> ELSE

-> SET result = FALSE;

-> SET msg = CONCAT('Product is ', product, ' which is less than 100.');

-> END IF;

-> -- Insert into product table

-> INSERT INTO product values(num1, num2, product, msg);

-> END;//

Query OK, 0 rows affected (0.01 sec)

mysql> DELIMITER ;

mysql> CALL CheckProduct(10.5, 9.6);

Query OK, 1 row affected (0.01 sec)

mysql> CALL CheckProduct(5.0, 6.0);

Query OK, 1 row affected (0.00 sec)

mysql> SELECT \* FROM PRODUCT;

+------+------+---------+---------------------------------------------------------+

| num1 | num2 | product | message |

+------+------+---------+---------------------------------------------------------+

| 10.5 | 9.6 | 100.8 | Product is 100.8 which is equal to or greater than 100. |

| 5 | 6 | 30 | Product is 30 which is less than 100. |

+------+------+---------+---------------------------------------------------------+

2 rows in set (0.00 sec)

