

# PL Exercise

## 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,  $\text{Perimeter} = 2 * (L + W)$  and  $\text{Area} = L * W$ .)

```
mysql> CREATE TABLE temp (AREA FLOAT, Perimeter FLOAT);
Query OK, 0 rows affected (0.05 sec)
```

```
mysql> DELIMITER //
mysql>
mysql> CREATE PROCEDURE RectangleCalc()
-> BEGIN
-> -- Declare variables
-> DECLARE length FLOAT DEFAULT 10;
-> DECLARE width FLOAT DEFAULT 5;
-> DECLARE area FLOAT;
-> DECLARE perimeter FLOAT;
->
-> -- Calculate area and perimeter
-> SET area = length * width;
-> SET perimeter = 2 * (length + width);
->
-> -- Output the results
-> INSERT INTO temp VALUES (area, perimeter);
-> END;
-> //
```

```
Query OK, 0 rows affected (0.00 sec)
```

```
mysql>
mysql> DELIMITER ;
mysql> Call RectangleCalc();
Query OK, 1 row affected (0.01 sec)
```

```
mysql> Select * from temp;
```

```
+-----+-----+
| AREA | Perimeter |
+-----+-----+
| 50 | 30 |
+-----+-----+
```

```
1 row in set (0.00 sec)
```

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

```
mysql> CREATE TABLE temp (Number INT, Square INT, `Cube` INT);
Query OK, 0 rows affected (0.03 sec)
```

```
mysql> DELIMITER //
mysql>
mysql> CREATE PROCEDURE ComputeNum()
-> BEGIN
-> -- Declare variables
-> DECLARE num INT DEFAULT 4;
-> DECLARE square INT;
-> DECLARE `cube` INT;
->
-> -- Calculate square and cube
-> SET square = num * num;
-> SET `cube` = num * num * num;
->
-> -- Insert the result into the table
-> INSERT INTO temp VALUES (num, square, `cube`);
-> END;
-> //
```

Query OK, 0 rows affected (0.01 sec)

```
mysql>
mysql> DELIMITER ;
mysql>
mysql> call computeNUM();
Query OK, 1 row affected (0.01 sec)
```

```
mysql> Select * from temp;
```

Number	Square	Cube
4	16	64

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> CREATE TABLE temperature_conversion (  
-> Fahrenheit FLOAT,  
-> Celsius FLOAT  
-> );
```

Query OK, 0 rows affected (0.89 sec)

```
mysql> DELIMITER //
```

```
mysql>
```

```
mysql> CREATE PROCEDURE ConvertTemperature()
```

```
-> BEGIN
```

```
-> -- Declare variables
```

```
-> DECLARE F FLOAT DEFAULT 98.6; -- example Fahrenheit
```

```
-> DECLARE C FLOAT DEFAULT 37.0; -- example Celsius
```

```
-> DECLARE convertedC FLOAT;
```

```
-> DECLARE convertedF FLOAT;
```

```
->
```

```
-> -- Perform conversions
```

```
-> SET convertedC = (F - 32) * 5 / 9;
```

```
-> SET convertedF = (9 / 5 * C) + 32;
```

```
->
```

```
-> -- Insert results into the table
```

```
-> INSERT INTO temperature_conversion (Fahrenheit, Celsius)
```

```
-> VALUES (F, convertedC), (convertedF, C);
```

```
-> END;
```

```
-> //
```

Query OK, 0 rows affected (0.01 sec)

```
mysql>
```

```
mysql> DELIMITER ;
```

```
mysql> CALL ConvertTemperature();
```

Query OK, 2 rows affected (0.02 sec)

```
mysql> SELECT * FROM temperature_conversion;
```

```
+-----+-----+  
| Fahrenheit | Celsius |  
+-----+-----+  
| 98.6 | 37 |  
| 98.6 | 37 |  
+-----+-----+
```

2 rows in set (0.01 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 inch_conversion (
```

```
-> TotalInches INT,  
-> Yards INT,  
-> Feet INT,  
-> Inches INT  
-> );
```

Query OK, 0 rows affected (0.70 sec)

```
mysql> DELIMITER //
```

```
mysql>
```

```
mysql> CREATE PROCEDURE ConvertInches()
```

```
-> BEGIN  
-> DECLARE totalInches INT DEFAULT 124;  
-> DECLARE remainingInches INT;  
-> DECLARE yards INT;  
-> DECLARE feet INT;  
-> DECLARE inches INT;  
->  
-> -- Calculate yards  
-> SET yards = totalInches DIV 36;  
-> SET remainingInches = totalInches MOD 36;  
->  
-> -- Calculate feet from remaining inches  
-> SET feet = remainingInches DIV 12;  
-> SET inches = remainingInches MOD 12;  
->  
-> -- Insert into the table  
-> INSERT INTO inch_conversion (TotalInches, Yards, Feet, Inches)  
-> VALUES (totalInches, yards, feet, inches);  
-> END;  
-> //
```

Query OK, 0 rows affected (0.53 sec)

```
mysql>
```

```
mysql> DELIMITER ;
```

```
mysql> CALL ConvertInches();
```

Query OK, 1 row affected (0.01 sec)

```
mysql> SELECT * FROM inch_conversion;
```

```
+-----+-----+-----+-----+  
| TotalInches | Yards | Feet | Inches |  
+-----+-----+-----+-----+  
|    124    |    3  |    1  |    4    |  
+-----+-----+-----+-----+
```

1 row in set (0.34 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> CREATE TABLE check_divisible_by_5 (  
->   Number INT,  
->   Result VARCHAR(50)  
-> );
```

Query OK, 0 rows affected (0.31 sec)

```
mysql> DELIMITER //  
mysql>
```

```
mysql> CREATE PROCEDURE CheckDivisibleBy5(IN inputNumber INT)  
-> BEGIN  
->   DECLARE message VARCHAR(50);  
->  
->   IF MOD(inputNumber, 5) = 0 THEN  
->     SET message = CONCAT(inputNumber, ' is divisible by 5');  
->   ELSE  
->     SET message = CONCAT(inputNumber, ' is NOT divisible by 5');  
->   END IF;  
->  
->   INSERT INTO check_divisible_by_5 (Number, Result)  
->   VALUES (inputNumber, message);  
-> END;  
-> //
```

Query OK, 0 rows affected (0.01 sec)

```
mysql>  
mysql> DELIMITER ;  
mysql> CALL CheckDivisibleBy5(25);  
Query OK, 1 row affected (0.01 sec)
```

```
mysql> CALL CheckDivisibleBy5(14);  
Query OK, 1 row affected (0.01 sec)
```

```
mysql> SELECT * FROM check_divisible_by_5;
```

Number	Result
25	25 is divisible by 5
14	14 is NOT divisible by 5

2 rows in set (0.00 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_check (  
->   Num1 FLOAT,  
->   Num2 FLOAT,  
->   Product FLOAT,  
->   Result VARCHAR(50)  
-> );
```

Query OK, 0 rows affected (0.14 sec)

```
mysql> DELIMITER //
```

```
mysql>
```

```
mysql> CREATE PROCEDURE CheckProduct(IN num1 FLOAT, IN num2 FLOAT)  
-> BEGIN  
->   DECLARE prod FLOAT;  
->   DECLARE message VARCHAR(50);  
->  
->   SET prod = num1 * num2;  
->  
->   IF prod >= 100 THEN  
->     SET message = 'Product is equal to or greater than 100';  
->   ELSE  
->     SET message = 'Product is less than 100';  
->   END IF;  
->  
->   INSERT INTO product_check (Num1, Num2, Product, Result)  
->     VALUES (num1, num2, prod, message);  
-> END;  
-> //
```

Query OK, 0 rows affected (0.01 sec)

```
mysql>
```

```
mysql> DELIMITER ;
```

```
mysql>
```

```
mysql> CALL CheckProduct(10.5, 9.7); -- Less than 100
```

Query OK, 1 row affected (0.01 sec)

```
mysql> CALL CheckProduct(20, 6); -- Equal to or greater than 100
```

Query OK, 1 row affected (0.00 sec)

```
mysql> SELECT * FROM product_check;
```

```
+-----+-----+-----+-----+  
| Num1 | Num2 | Product | Result |  
+-----+-----+-----+-----+  
| 10.5 | 9.7 | 101.85 | Product is equal to or greater than 100 |  
| 20 | 6 | 120 | Product is equal to or greater than 100 |  
+-----+-----+-----+-----+
```

2 rows in set (0.00 sec)