

## FUNCIONES PL/SQL

Escribir una función en pl/sql con las siguientes características:

Nombre: fn\_addition\_01

Parámetros: valor\_inferior, valor\_superior (p\_min, p\_max)

Tipo de dato de retorno: int

Descripción: Realizar la suma de p\_min y p\_max

Ejemplo: Para los valores de entrada 1 y 10 el cálculo sería  $1+10 = 11$

```
MariaDB [world]> delimiter $$
MariaDB [world]> CREATE FUNCTION fn_addition_01(
    -> p_min int,
    -> p_max int
    -> )
    -> returns int
    -> deterministic
    -> begin
    -> declare v_result int;
    ->
    -> set v_result := p_min + p_max;
    ->
    -> return v_result;
    -> end;
    -> $$
Query OK, 0 rows affected (0.006 sec)

MariaDB [world]> SELECT fn_addition_01(1,10)$$
+-----+
| fn_addition_01(1,10) |
+-----+
|                  11 |
+-----+
1 row in set (0.001 sec)
```

Escribir una función en pl/sql con las siguientes características:

Nombre: fn\_triangle\_area

Parámetros: lado, altura (p\_side, p\_height)

Tipo de dato de retorno: real

Descripción: Calcular el área de un triángulo, la cual se hace con la fórmula  
(base \* altura) / 2

Ejemplo: Para los valores de entrada 3 y 5, el valor de retorno es 7.5

Query OK, 0 rows affected (0.007 sec)

1 row in set (0.001 sec)

Nombre: fn arithmetic operation 01

Tipo de dato de retorno: real

números p\_val1 y p\_val2. Los posibles valores para p\_operation son:

Ejemplo: Para los valores de entrada 1, 10, "addition" debe retornar 11. Para los valores 10, 5, "subtraction" debe retornar 5.

```
Query OK, 0 rows affected (0.006 sec)
```

```

+-----+
1 row in set (0.002 sec)

```

```
+-----+
1 row in set (0.001 sec)
```

Escribir una función en pl/sql con las siguientes características:

Nombre: fn\_number\_type\_01

Parámetros: número (p\_number)

Tipo de dato de retorno: varchar

Descripción: Si el número de entrada es par debe retornar "EVEN", en caso contrario debe retornar "ODD".

Ejemplo: Para el valor 2 debe retornar EVEN. Para el valor 3, ODD.

```
MariaDB [world]> delimiter $$
MariaDB [world]> create function fn_number_type_01(
  -> number int
  -> )
  -> returns VARCHAR(10)
  -> deterministic
  -> begin
  ->     if (number % 2 = 0) then
  ->         return 'EVEN';
  ->     else
  ->         return 'ODD';
  ->     end if;
  -> end;
  -> $$
```

**Query OK, 0 rows affected (0.005 sec)**

```
MariaDB [world]> SELECT fn_number_type_01(3)$$
```

```
+-----+
| fn_number_type_01(3) |
+-----+
| ODD                  |
+-----+
```

**1 row in set (0.001 sec)**

```
MariaDB [world]> SELECT fn_number_type_01(2)$$
```

```
+-----+
| fn_number_type_01(2) |
+-----+
| EVEN                  |
+-----+
```

**1 row in set (0.001 sec)**

Ejemplo: Para los valores de entrada 3 y 5, el valor de retorno es NO

```

MariaDB [world]> CREATE FUNCTION fn_is_triangle_area_even_01(
-> p_side real,
-> p_height real
-> )
-> returns VARCHAR(10)
-> deterministic
-> begin
->     declare v_triangle_area real;
->     declare v_even_check VARCHAR(10);
->
->     set v_triangle_area := fn_triangle_area(p_side,p_height);
->     set v_even_check := fn_number_type_01(v_triangle_area);
->
->     if (v_even_check = 'EVEN') then
->         return 'YES';
->     else
->         return 'NO';
->     end if;
-> end;
-> $$
ERROR 1106 (HY000): Unknown system variable 'v_triangle_area'
MariaDB [world]> CREATE FUNCTION fn_is_triangle_area_even_01( p_side real, p_height real ) returns VARCHAR(10) deterministic begin
triangle_area(p_side,p_height); set v_even_check := fn_number_type_01(v_triangle_area); if (v_even_check = 'EVEN') then
declare v_triangle_area real; declare v_even_check VARCHAR(10); set v_triangle_area := fn_
return 'YES'; else return 'NO'; end if; end;$$
Query OK, 0 rows affected (0.007 sec)

```

```
1 row in set (0.002 sec)
```