Create function

CREATE FUNCTION *sp\_name* ([*parameter*[,...]])

RETURNS *type*

[*characteristic* ...] *routine\_body*

*parameter*:

[ IN | OUT | INOUT ] *param\_name* *type*

*type*:

*Any valid MySQL data type*

*characteristic*:

LANGUAGE SQL

| [NOT] DETERMINISTIC

| { CONTAINS SQL | NO SQL | READS SQL DATA | MODIFIES SQL DATA }

| SQL SECURITY { DEFINER | INVOKER }

| COMMENT '*string*'

*routine\_body*:

*procedimientos almacenados o comandos SQL válidos*

**Sentencia compuesta BEGIN ... END**

[*begin\_label*:] BEGIN

[*statement\_list*]

END [*end\_label*]

### Sentencia DECLARE

DECLARE *var\_name*[,...] *type* [DEFAULT *expresion*]

Este comando se usa para declarar variables locales. Para proporcionar un valor por defecto para la variable, incluya una cláusula **DEFAULT** . El valor puede especificarse como expresión, no necesita ser una constante. Si la cláusula **DEFAULT** no está presente, el valor inicial es **NULL**.

La visibilidad de una variable local es dentro del bloque **BEGIN ... END** donde está declarado. Puede usarse en bloques anidados excepto aquéllos que declaren una variable con el mismo nombre.

#### Sentencia SET para variables

SET *var\_name* = *expr* [, *var\_name* = *expr*] ...

El comando **SET** en funciones es una versión extendida del comando general **SET**. Las variables referenciadas pueden ser las declaradas dentro de una rutina, o variables de servidor globales.

#### Sentencia IF

IF *search\_condition* THEN *statement\_list*

[ELSEIF *search\_condition* THEN *statement\_list*] ...

[ELSE *statement\_list*]

END IF

**IF** implementa un constructor condicional básico. Si *search\_condition* se evalúa a cierto, el comando SQL correspondiente listado se ejectua. Si no coincide ninguna *search\_condition* se ejecuta el comando listado en la cláusula **ELSE**. *statement\_list* puede consistir en varios comandos.

#### La sentencia CASE

CASE *case\_value*

WHEN *when\_value* THEN *statement\_list*

[WHEN *when\_value* THEN *statement\_list*] ...

[ELSE *statement\_list*]

END CASE

O:

CASE

WHEN *search\_condition* THEN *statement\_list*

[WHEN *search\_condition* THEN *statement\_list*] ...

[ELSE *statement\_list*]

END CASE

El comando **CASE**  implementa un constructor condicional complejo. Si una *search\_condition* se evalúa a cierto, el comando SQL correspondiente se ejecuta. Si no coincide ninguna condición de búsqueda, el comando en la cláusula **ELSE** se ejecuta.

#### Sentencia LOOP

[*begin\_label*:] LOOP

*statement\_list*

END LOOP [*end\_label*]

**LOOP** implementa un constructor de bucle simple que permite ejecución repetida de comandos particulares. El comando dentro del bucle se repite hasta que acaba el bucle, usualmente con un comando **LEAVE** .

#### Sentencia LEAVE

LEAVE *label*

Este comando se usa para abandonar cualquier control de flujo etiquetado. Puede usarse con **BEGIN ... END** o bucles.

#### La setencia ITERATE

ITERATE *label*

**ITERATE** sólo puede aparecer en comandos **LOOP**, **REPEAT**, y **WHILE** . **ITERATE** significa “vuelve a hacer el bucle.”

Por ejemplo:

CREATE PROCEDURE doiterate(p1 INT)

BEGIN

label1: LOOP

SET p1 = p1 + 1;

IF p1 < 10 THEN ITERATE label1; END IF;

LEAVE label1;

END LOOP label1;

SET @x = p1;

END

#### Sentencia REPEAT

[*begin\_label*:] REPEAT

*statement\_list*

UNTIL *search\_condition*

END REPEAT [*end\_label*]

El comando/s dentro de un comando **REPEAT** se repite hasta que la condición *search\_condition* es cierta.

Un comando **REPEAT** puede etiquetarse. *end\_label* no puede darse a no ser que *begin\_label* esté presente, y si lo están, deben ser el mismo.

#### Sentencia WHILE

[*begin\_label*:] WHILE *search\_condition* DO

*statement\_list*

END WHILE [*end\_label*]

El comado/s dentro de un comando **WHILE** se repite mientras la condición *search\_condition* es cierta.

Un comando **WHILE** puede etiquetarse. *end\_label* no puede darse a no ser que *begin\_label*también esté presente, y si lo están, deben ser el mismo.

Por ejemplo:

CREATE PROCEDURE dowhile()

BEGIN

DECLARE v1 INT DEFAULT 5;

WHILE v1 > 0 DO

...

SET v1 = v1 - 1;

END WHILE;

END

Ejercicios:

1. Formato fecha
2. Hora: Son las….
3. Capital
4. Camel