Funciones

Tipos de variables en MySql

- Variables globales del sistema: definen comportamiento global para todas las conexiones.
  - Cuando arrancamos el servidor éste inicializa el valor de las variables según el archivo my.cnf o en los argumentos del comando mysqld o sus valores por defecto.
  - Podemos modificar el valor de una variable global con el siguiente comando:

```
SET GLOBAL nombre_variable = nuevo_valor;
```

  - Por compatibilidad con otros sistemas, MySql permite referirnos a las variables globales con los modificadores `@@`. Así por ejemplo, la expresión anterior quedaría:

```
SET @@global.nombre_variable = nuevo_valor
```

  - Para poder cambiar una variable global debemos disponer de permisos de SUPER.

- Variables de sesión. Para cada conexión MySql genera lo que se llaman variables de sesión. Para obtenerlas, simplemente las copia de las variables globales.
  - Cualquier usuario puede cambiar sus variables de sessión. No necesita permisos. Eso sí, sólo puede cambiar las de su sesión.
  - Para cambiar el valor, utilizamos:
  - Para referirnos a una v “@@local” o, directamente, “@@”.

```
SET nombre_variable = nuevo valor;
```

  - Cuando cambiamos el valor de una variable de sesión, dicho valor permanece mientras dura la sesión.
  - Para visualizar el valor de las variables de sesión: “SHOW variables”. Incluso si utilizamos “SHOW GLOBAL variables” siempre nos mostrará las variables de sesión.

- Variables de usuario.
  - En cualquier sesión se pueden alojar en el servidor variables. Tendrán validez sólo durante la sesión, por lo tanto, no podrán ser usadas por otros clientes.
  - Se nombrarán @nombre_variable. Donde nombre_variable podrá contener cualquier valor alfanumérico, el punto, el guión bajo y el símbolo del dólar.
  - Una forma de crear variables de usuario es:

```
SET @nombre_variable=expresión [, @nombre_variable2=expresión2]...
```

  - Otra forma es con SELECT pero en este caso utilizando “:=”:

```
SELECT nombre_variable := expresión...;
```

  - Una variable no inicializada tiene un valor NULL.

Funciones

CREATE [DEFINER = { user | CURRENT_USER }] FUNCTION sp_name ([func_parameter[,...]]) RETURNS type [characteristic ...] routine_body

func_parameter: param_name type

type:Any valid MySQL data type

characteristic:COMMENT 'string'| LANGUAGE SQL| [NOT] DETERMINISTIC

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

| SQL SECURITY { DEFINER | INVOKER }

```
```

- Una función está asociada a una base de datos, igual que un procedimiento.
- Se invoca utilizando nombre_función (parámetros). Exactamente igual que las funciones ya preestablecidas como CURDATE() y demás.
- Todos los parámetros son IN.
- La cuestión de los privilegios es exactamente igual que en el caso de los procedimientos.
- La cláusula RETURNS es obligatoria y el cuerpo de la rutina debe contener la instrucción RETURN valor. El tipo de uno y otro debe coincidir si no, se pueden producir efectos indeseados.
- Funciones no pueden contener: COMMIT, ROLLBACK, DESCRIBE, EXPLAIN, SELECT sin cláusula INTO, USE y, en general, cualquier instrucción que devuelva un conjunto de valores al cliente.
- Las funciones no pueden ser recursivas. (Los procedimientos sí, pero está deshabilitado por defecto).

Control de flujo

Las diferentes instrucciones de control de flujo las iremos viendo en las clases prácticas. Aquí, como muestra, un botón:

```
IF search_condition THEN statement_list
  [ELSEIF search_condition THEN statement_list] ...
  [ELSE statement_list]  END IF;
```

- Si la expresión <search_condition> es cierta, se ejecuta la primera lista de instrucciones, si no, se ejecuta las instrucciones situadas después del ELSE.
- Puede haber anidación de unos IFs con otros pero siempre acabarán con punto y coma después del END IF.

Problemas: Funciones_1

1. Escribe una función que devuelva tu nombre.
2. Escribe una función que reciba un parámetro de tipo varchar y devuelva ese mismo parámetro.
3. Escribe una función que reciba un varchar como parámetro de entrada y devuelva la segunda letra del varchar recibido.
4. Escribe una función que devuelva el JEFE del nombre del empleado pasado como parámetro de entrada. (tabla emple)
5. Escribe una función que devuelva una fecha dada con el formato: día de la semana, día del mes, hora, minutos y segundos.
6. Escribe una función que devuelva el número de años completos de diferencia entre dos fechas que se le pasan como parámetros.

Problemas: Funciones_2

1. Desarrolla una función que devuelva el precio total de una comanda de la bbdd empresa.
2. Escribe una función que devuelva la dirección postal completa de un cliente que le pasamos como parámetro de entrada.
3. Escribe una función que compare dos letras minúsculas del rango [a..z] que le pasamos como parámetros de entrada y que, a continuación, imprima la mayor de las dos o cero en el caso de que sean iguales.