

Bases de Datos

Unidad 8: Programación de bases de datos Sesión 5

3.- Desarrollo de funciones

Las funciones son rutinas compuestas por varias instrucciones SQL que devuelven un resultado. **Respecto de los procedimientos, las funciones presentan las siguientes diferencias:**

- ☐ Las funciones **devuelven siempre un dato** a través de una instrucción **RETURN**. El dato se corresponde con un tipo declarado para la función.
- ☐ Las funciones no pueden trabajar con parámetros OUT o INOUT.
- ☐ Las funciones son llamadas a **ejecución**, al igual que las funciones propias de MySQL, **escribiendo su nombre y la lista de parámetros pasados a la función encerrados entre paréntesis**. Por tanto, no usa una instrucción de llamada como la instrucción CALL de llamada a los procedimientos.
- ☐ Las funciones podrán ser llamadas desde cualquier instrucción SQL como SELECT, UPDATE, INSERT, DELETE. Los procedimientos nunca pueden ser llamados a ejecución dentro de otra instrucción.
- ☐ **En una función no se puede usar SELECT**, salvo cuando lo devuelto se asigna en una variable.

3.- Desarrollo de funciones

Sintaxis para crear una función:

```
CREATE FUNCTION nomFuncion([parametro[,...]])  
  RETURNS tipo  
  [característica ...]  
BEGIN  
      CuerpoRutina  
END
```

Con respecto a los procedimientos, es nueva la cláusula **RETURNS tipo** que sirve para indicar el tipo de dato resultado que devuelve la función.

Para devolver un resultado, la función debe incluir dentro del cuerpo de la rutina, la instrucción **RETURN expresion**, debiendo ser expresión del mismo tipo que la función.

3.- Desarrollo de funciones

Ejemplo 1: Realiza una función que devuelve si un número entero es par o impar.

```
CREATE FUNCTION par (n INT)
RETURNS BOOLEAN
BEGIN
    if n%2=0 then
        RETURN true;
    else
        RETURN false;
    end if;
END ;
```

Ejemplos de llamada a la función par:

SELECT par(7);

SET @p=par(7);

SET @p=par((select count() from automoviles));*

SELECT count() from contratos where par(numcontrato)=true;*

3.- Desarrollo de funciones

Ejemplo 2: Realiza una función para obtener cuantos divisores tiene un número entero positivo. Si el número es cero, devolverá que tiene cero divisores.

```
CREATE FUNCTION divisores (n INT)
RETURNS INT
BEGIN
    DECLARE d INT DEFAULT 0;
    DECLARE c INT DEFAULT 0;
    IF n < 0 THEN
        SET n = -n;
    END IF;
    IF n <> 0 THEN
        WHILE d < n DO
            SET d = d + 1;
            IF n % d = 0 THEN
                SET c = c + 1;
            END IF;
        END WHILE;
    END IF;
    RETURN c;
END;
```

3.- Desarrollo de funciones

Ejemplo 3: Realiza una función para obtener si un número entero es o no es primo.

```
CREATE FUNCTION es_primo (n INT)
RETURNS BOOLEAN
BEGIN
    DECLARE es BOOLEAN;
    DECLARE nd int;
    IF n>0 THEN
        SET es=false;
        SET nd=divisores(n);
        IF (nd=2 OR nd=1) THEN
            SET es=true;
        END IF;
    END IF;
    RETURN es;
END;
```