**MÁS EJERCICIOS FUNCIONES Y PROCEDIMIENTOS**

1. Crea dentro del esquema prueba una tabla llamada tabla2 con dos campos (uno de tipo entero y otro de tipo VARCHAR(10)). A continuación, crea un procedimiento que añadirá a la tabla los siguientes valores: 1 y dato1, 2 y dato2….10 y dato10.

Escribid el procedimiento y mostrad el resultado obtenido al ejecutar.

CREATE PROCEDURE `crearTabla2` ()

BEGIN

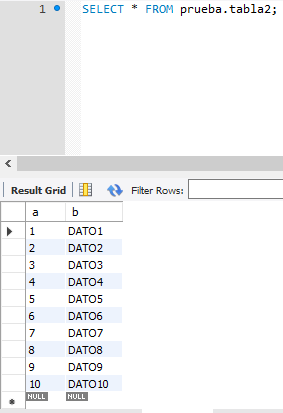
declare contador int default 1;

while contador <= 10 do

insert into tabla2 values (contador, concat("DATO",contador));

set contador = contador + 1;

end while;

END

1. Crea dentro del esquema prueba una tabla llamada tabla3 con dos campos uno de tipo INT y otro de tipo VARCHAR(30). No hace falta que pongas clave. A continuación, crea un procedimiento que recibirá un número entero como parámetro. Dicho procedimiento introducirá en la tabla el número y la cadena “MAYOR DE 3” si el número es mayor de 3, el número y “MENOR DE 3” si es menor y el número e “IGUAL a 3” si es igual. El número lo introduciremos como parámetro.

Escribid el procedimiento y mostrad el resultado obtenido al ejecutar.

CREATE PROCEDURE `crearTabla3` (x int)

BEGIN

declare resultado varchar(30) default " es igual que 3";

if x > 3 then

set resultado = " es mayor que 3";

end if;

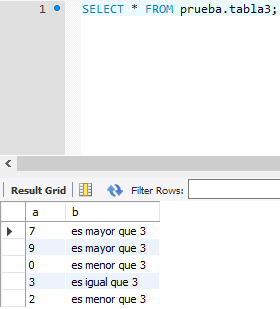
if x < 3 then

set resultado = " es menor que 3";

end if;

insert into tabla3 values (x, resultado);

END;



1. Crea un procedimiento en el esquema prueba el cual, utilizando la función que devolvía el nombre del día de la semana, me devuelva el nombre de los 7 días. Haz lo mismo con otro procedimiento, pero, en este caso, que devuelva el nombre de los 12 meses del año usando la función correspondiente.

En cada una de las 7 pestañas saldrá un día de la semana.

Con los meses, la misma operación.

**Días:**

CREATE FUNCTION `DiasSemana`(x INT)

RETURNS varchar(15)

BEGIN

DECLARE a VARCHAR(15);

CASE x

when 1 then set a="LUNES";

when 2 then set a="MARTES";

when 3 then set a="MIÉRCOLES";

when 4 then set a="JUEVES";

when 5 then set a="VIERNES";

when 6 then set a="SÁBADO";

when 7 then set a="DOMINGO";

ELSE SET a="ERROR";

END CASE;

RETURN a;

END

**-------------------------------------------------------------------------**

CREATE PROCEDURE `crearTabla5` ()

BEGIN

declare contador int default 1;

while contador <= 7 do

insert into tabla5 select prueba.DiasSemana(contador);

set contador = contador + 1;

end while;

END

**Meses:**

CREATE FUNCTION `MesAno`(x INT)

RETURNS varchar(20)

BEGIN

DECLARE a VARCHAR(20);

CASE x

when 1 then set a="ENERO";

when 2 then set a="FEBRERO";

when 3 then set a="MARZO";

when 4 then set a="ABRIL";

when 5 then set a="MAYO";

when 6 then set a="JUNIO";

when 7 then set a="JULIO";

when 8 then set a="AGOSTO";

when 9 then set a="SEPTIEMBRE";

when 10 then set a="OCTUBRE";

when 11 then set a="NOVIEMBRE";

when 12 then set a="DICIEMBRE";

ELSE SET a="ERROR";

END CASE;

RETURN a;

END

**-------------------------------------------------------------------------**

CREATE PROCEDURE `crearTabla4` ()

BEGIN

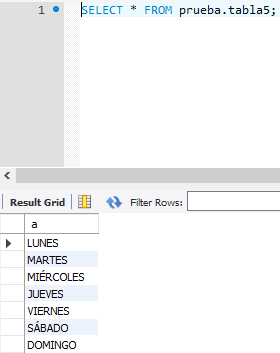
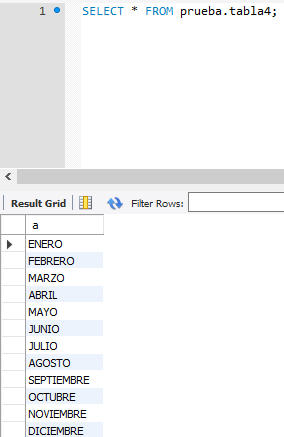
declare contador int default 1;

while contador <= 12 do

insert into tabla4 select prueba.MesAno(contador);

set contador = contador + 1;

end while;

END

1. Crea una función llamada IVA dentro del esquema prueba. Esta función recibirá un parámetro de tipo FLOAT (porque podrá tener decimales) y aplicará a dicho parámetro un IVA del 21%. La función calculará el valor que habrá que sumar al posteriormente al parámetro.

Ejemplo: Si luego llamo a la función SELECT IVA(100), la función devolverá 21. El precio final sería el resultado de sumar a 100 los 21.

A continuación, utilízala para mostrar el total de ventas en euros para cada uno de los productos del esquema ventas, una vez aplicada dicha función.

CREATE FUNCTION `IVA` (x float)

RETURNS FLOAT

BEGIN

declare aDevolver float default 0;

if x < 0 then

set aDevolver = 0;

else

set aDevolver = x\*0.21;

end if;

RETURN aDevolver;

END

**-------------------------------------------------------------------------**

SELECT \*, prueba.IVA(precio) + precio as conIVA FROM ventas.productos;

1. Modifica la función anterior para que introduzcamos 2 parámetros, el primero será el valor sobre el que voy a aplicar el IVA (de tipo FLOAT como antes) y el segundo será el tipo de IVA a aplicar (de tipo INT)

El primer valor es el parámetro sobre el que se aplicará el IVA. El segundo parámetro indica el porcentaje de IVA que se va a aplicar.

Es decir, en este caso, la función nos devolverá 18.

Una vez creada, aplícala para calcular lo que te mandaba el ejercicio anterior en el esquema ventas.

CREATE FUNCTION `IVA` (valor float, iva int)

RETURNS FLOAT

BEGIN

declare aDevolver float default 0;

if valor < 0 then

set aDevolver = 0;

else

set aDevolver = (valor\*iva)/100;

end if;

RETURN aDevolver;

END

**-------------------------------------------------------------------------**

SELECT \*, prueba.IVA(21, precio) + precio as conIVA FROM ventas.productos;

1. Dentro del esquema supermercado, muestra para cada uno de los vendedores su **nombre completo**, su **edad** (recuerda que tienes una función años que te la devuelve) y el **nombre del mes en el que ha sido dado de alta** (en español), pero sólo para el caso de aquellos vendedores cuyo mes en el que fue dado de alta **empiece por m**. Utiliza la función que devolvía el nombre del mes.

**UTILIZA LAS DOS FUNCIONES DE LAS QUE TE HABLO. PARA ELLO, ES NECESARIO TENERLAS CREADAS EN LOS EJERCICIOS PREVIOS. NO ME VALE QUE LO HAGÁIS DE LA OTRA FORMA, PORQUE LO QUE QUIERO ES QUE PRACTIQUÉIS CON FUNCIONES CREADAS POR VOSOTROS.**

SELECT concat(nombre, " ", apellidos) as Nombre, ventas.años(fechanacim) as Edad, prueba.MesAno(month(fechaalta)) as MesAlta FROM supermercado.vendedores having MesAlta like ("M%");