**EJERCICIOS REPASO**

1. Vas a crear un procedimiento en el esquema mundo. Lo primero que has de hacer es crear una nueva tabla llamada NUEVA que contendrá 4 campos de tipo VARCHAR(40). Dicha tabla contendrá los campos NUEVOPAÍS, NUEVACAPITAL, NUEVA MONEDA Y NUEVALENGUA.

El procedimiento recibirá un continente como parámetro y se ocupará de guardar en la tabla NUEVA los datos de los países que pertenecen a ese continente, pero sólo aquellos que hablen inglés. A continuación, y dentro del procedimiento, muestra el contenido de los datos añadidos a NUEVA.

**(2,5 PUNTOS)**

CREATE DEFINER=`root`@`localhost` PROCEDURE `paises\_continente`(conti varchar(30))

BEGIN

declare paisx varchar(40);

declare capitalx varchar(40);

declare monedax varchar(40);

declare lenguax varchar(40);

declare recorrido boolean;

declare cursor1 cursor FOR SELECT pais, capital, moneda, lengua from paises where continente = conti and lengua = "Inglés";

declare CONTINUE HANDLER FOR NOT FOUND SET recorrido=1;

set recorrido = 0;

open cursor1;

while recorrido = 0 do

fetch cursor1 INTO paisx, capitalx, monedax, lenguax;

if recorrido=0 then

insert into nueva values(paisx, capitalx, monedax, lenguax);

select paisx, capitalx, monedax, lenguax;

end if;

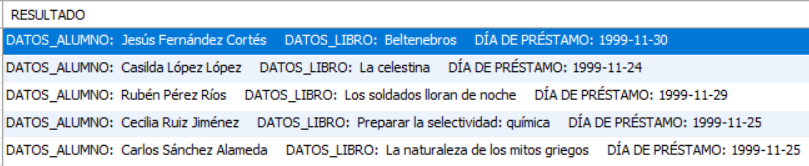
end while;

END

He supuesto que *“muestra el contenido de los datos añadidos a NUEVA” significa mostrar los datos por separado con el select del final.*

1. Crea un procedimiento en el esquema biblioteca. El procedimiento recibirá un nombre de mes (en español) y devolverá en un único campo el nombre y apellidos de los alumnos que han prestamos libros en ese mes, el título de los libros prestados y la fecha de préstamo, pero sólo para aquellos libros que no hayan sido devueltos. Ordena el listado por apellidos de los alumnos.

La salida de resultados aparecerá de la siguiente forma:

****

Recuerda que tienes una función que me devuelve el nombre del mes en español.

**(3,5 PUNTOS)**

CREATE DEFINER=`root`@`localhost` PROCEDURE `consultaPorMes`(mesConsulta varchar(30))

BEGIN

select concat("DATOS\_ALUMNO: ", alumnos.Nombre, " ", alumnos.apellidos, " DATOS\_LIBRO: ", libros.titulo, " DÍA DE PRÉSTAMOS: ", prestamos.fechaprestamo) as RESULTADO

from libros, alumnos, prestamos

where libros.registro = prestamos.codigolibro and prestamos.ncarnet = alumnos.ncarnet and

prestamos.fechadevolucion is null and prueba.MesAno(month(prestamos.fechaprestamo)) = mesConsulta

order by alumnos.apellidos;

END

1. Crea un trigger dentro de la tabla nuevaarticulos del esquema supermercado, la cual será una copia de la tabla artículos. El trigger va a controlar la inserción de datos en dicha tabla. **(4 PUNTOS)**
   1. En caso de que se introduzca un número de departamento que no exista, se asignará como código de departamento un 1 y si introducimos como precio del artículo un número menor o igual a 0 cuando el departamento no existe asignaremos al nuevo precio 10 €.
   2. En caso de que el departamento sí exista, si el nuevo precio es <=0, se asignará a dicho artículo el precio mínimo de los artículos de su departamento.

CREATE DEFINER=`root`@`localhost` TRIGGER `nuevaarticulos\_BEFORE\_INSERT` BEFORE INSERT ON `nuevaarticulos` FOR EACH ROW BEGIN

if not exists (select dpto from nuevaarticulos where dpto = new.dpto) then

set new.dpto = 1;

if new.preciounidad <= 0 then

set new.preciounidad = 10;

end if;

else

if new.preciounidad <= 0 then

set new.preciounidad = (select min(preciounidad) from nuevaarticulos where dpto = new.dpto);

end if;

end if;

END