**MÁS 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.

**CREATE DEFINER=`root`@`localhost` PROCEDURE `conti`(x varchar(25))**

**BEGIN**

**IF X IN (SELECT DISTINCT CONTINENTE FROM PAISES) THEN**

**INSERT INTO NUEVA**

**SELECT PAIS, CAPITAL, MONEDA, LENGUA FROM PAISES**

**WHERE CONTINENTE=X AND LENGUA="INGLÉS";**

**select \* from nueva;**

**ELSE SELECT "ERROR";**

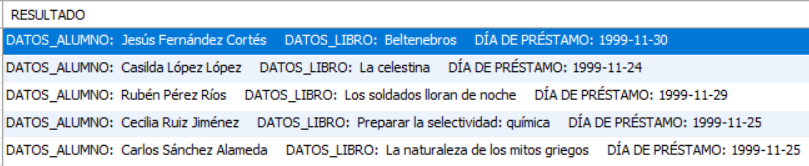
**end if;**

**END**

**YO OS HE HECHO LA COMPROBACIÓN DEL PARÁMETRO DE ENTRADA. NO OS LO PEDÍA EL EJERCICIO. ESTÁ MARCADO EN AZUL.**

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 prestado 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:

****

**CREATE DEFINER=`root`@`localhost` PROCEDURE `new\_procedure`(X varchar(30))**

**BEGIN**

**SELECT CONCAT\_WS(" ","DATOS\_ALUMNO: ",ALUMNOS.NOMBRE,ALUMNOS.APELLIDOS," DATOS\_LIBRO: ", LIBROS.TITULO, " DÍA DE PRÉSTAMO:",prestamos.fechaprestamo) AS RESULTADO FROM LIBROS INNER JOIN PRESTAMOS ON PRESTAMOS.CODIGOLIBRO=LIBROS.REGISTRO INNER JOIN ALUMNOS ON ALUMNOS.NCARNET=PRESTAMOS.NCARNET WHERE prueba.diames(month(FECHAPRESTAMO))=X AND FECHADEVOLUCION IS NULL order by 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 `supermercado`.`nuevaarticulos\_BEFORE\_INSERT` BEFORE INSERT ON `nuevaarticulos` FOR EACH ROW**

**BEGIN**

**declare x int;**

**select depart into x from departamentos where new.dpto=depart;**

**if x is not null then**

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

**else**

**set new.dpto=1;**

**if new.preciounidad<=0 then**

**set new.preciounidad=10;**

**end if;**

**end if;**

**END**