

Actividad de aprendizaje 8-02

Se trata de:

- Aprender a crear y ejecutar procedimientos de usuario.

Objetivos:

- Editar en Workbench un procedimiento.
- Almacenar un procedimiento en servidor.
- Ejecutar un procedimiento.
- Editar en cliente **mysql** un procedimiento.

Procedimiento

Debes importar la base de datos **datosorigen** cuyo archivo está adjunto con la tarea.

Esta base de datos contiene varias tablas sin relacionar que tienen datos fuente a partir de los cuales generaremos datos para otras tablas (preferentemente de personas). En esta base de datos:

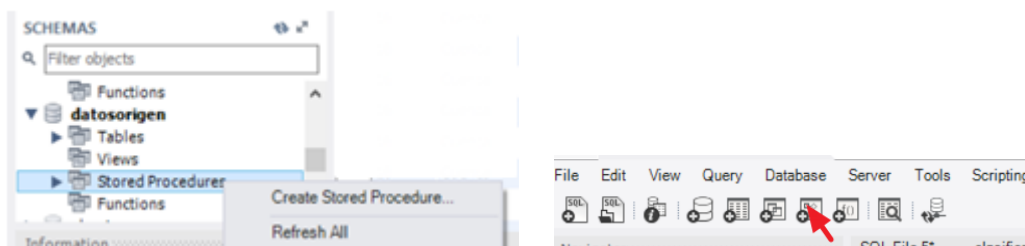
- La tabla **nombreshombre** contiene los 5000 nombres de hombre más frecuentes en España, con la frecuencia con la que se repiten. Están ordenados de mayor a menor frecuencia. Si te fijas, en todos los nombres hay una columna desde y una columna hasta. En esta tabla el valor hasta del último nombre es 126228. Esto querría decir que sobre un total de 126228 personas, en frecnombre se tendrían las frecuencias con las que se repite cada nombre y entre los valores de las columnas desde y hasta se tendrían todas las veces que se repetiría cada uno de esos nombres.
- La tabla **nombresmujer** contiene los 5000 nombres de mujer más frecuentes en España.
- La tabla **apellidos** contiene los 5000 apellidos más frecuentes en España.
- La tabla **municipios** contiene los datos de todos los municipios de España. Si todos los habitantes de España tuvieran un número autoincrementado y se ordenarán por municipios, entre los valores de las columnas desde y hasta se tendrían los números que identificarían a cada uno de los habitantes de un municipio.
- La tabla **municipioscantabria** contiene los datos de todos los municipios de Cantabria.
- La tabla **calles** contiene nombres de calles, avenidas, plazas, etc.

1.- Ahora vas a crear o editar un procedimiento desde Workbench, dentro de la base de datos **datororigen**, que:

- Crea, si es que no existe ya, una tabla de personas con las columnas num, nombre, apellido1, apellido 2, dni, fecha_nac, calle, localidad, sexo (H o M).
- Añade n filas a la tabla personas con un nombre obtenido aleatoriamente de la tabla de nombres de hombre o de la tabla de nombres de mujer.
- Devuelve el último número de persona que se ha generado al insertar las nuevas filas.

Para realizar los tres requerimientos anteriores deberás seguir los pasos que se indican a continuación:

1.1.- Solicita editar un nuevo procedimiento almacenado, en la opción correspondiente del menú contextual de objetos **Stored Procedures** o pulsando el botón **Create a new stored Procedure**.



1.2.- Se abre una ventana de edición con el esqueleto de un procedimiento.

```
CREATE PROCEDURE `new_procedure` ()  
BEGIN  
  
END
```

1.3.- Da el nombre **generar_nombres** al procedimiento.

```
CREATE PROCEDURE generar_nombres ()  
BEGIN  
  
END
```

1.4.- Establece que el procedimiento reciba el número de filas que queremos insertar y que devuelva el número de la última que se ha insertado.

```
CREATE PROCEDURE generar_nombres (IN n INT,OUT ultimo INT)
```

1.5.- Edita el código del procedimiento.

```
CREATE PROCEDURE generar_nombres(IN n INT,OUT ultimo INT)
BEGIN
  DECLARE i INT;
  DECLARE nmaxhombre FLOAT;
  DECLARE nmaxmujer FLOAT;
  DECLARE aleat1 FLOAT;
  DECLARE aleat2 INT;
  DECLARE nomaleat VARCHAR(25);

  CREATE TABLE IF NOT EXISTS personas (
    num INT PRIMARY KEY AUTO_INCREMENT,
    nombre VARCHAR(25),
    apellido1 VARCHAR(25),
    apellido2 VARCHAR(25),
    dni CHAR(9),
    fecha_nac DATE,
    direccion VARCHAR(50),
    localidad VARCHAR(70),
    sexo CHAR(1));

  SELECT max(hasta) INTO nmaxhombre FROM nombreshombre;
  SELECT max(hasta) INTO nmaxmujer FROM nombresmujer;

  SET i=0;
  WHILE i<n DO
    SET aleat1=rand();
    IF aleat1<0.5 THEN
      SET aleat2=floor(rand()*nmaxmujer);
      SELECT nombre INTO nomaleat FROM nombresmujer WHERE
        aleat>=desde AND aleat<hasta;
      INSERT INTO personas(nombre, sexo) VALUES (nomaleat,'M');
    ELSE
      SET aleat2=floor(rand()*nmaxhombre);
      SELECT nombre INTO nomaleat FROM nombreshombre WHERE
        aleat>=desde AND aleat<hasta;
      INSERT INTO personas(nombre, sexo) VALUES (nomaleat,'M');
    END IF;
    SET i=i+1;
  END WHILE;
  SET ultimo=(SELECT max(num) FROM personas);

END
```

2.- Aplica los cambios para que se envíe el código al servidor, que analizará el código y, si es correcto, lo **almacenará** para su posible ejecución. Observa que en el código que se envía se añaden algunas líneas a lo editado.

```
1  USE `datosorigen`;  
2  DROP procedure IF EXISTS `generar_nombres`;  
3  
4  DELIMITER $$  
5  USE `datosorigen` $$  
6  CREATE PROCEDURE generar_nombres (IN n INT,OUT ultimo INT)  
7  BEGIN  
8    DECLARE i INT;  
9    DECLARE nmaxhombre FLOAT;  
10   DECLARE nmaxmujer FLOAT;  
11   DECLARE aleat FLOAT;  
12   DECLARE nomaleat VARCHAR(25);  
13  
14   CREATE TABLE IF NOT EXISTS personas (  
15     num INT PRIMARY KEY AUTO_INCREMENT,  
16     nombre VARCHAR(25),  
17     apellido1 VARCHAR(25),  
18     apellido2 VARCHAR(25),
```

3.- Ejecuta el procedimiento desde línea de comandos para que inserte 100 nombres de personas y devuelva el último número de persona introducida en una variable @numper:

CALL generar_nombres(100,@numper);

Verás que al ejecutar se detecta un error en el código almacenado, no existe la variable **aleat**. Cambia las referencias a esta variable por **aleat2**. Vuelve a enviar a servidor el código del procedimiento.

Vuelve a ejecutar el procedimiento.

Obtén el último número de persona introducida.

SELECT @numper;

4.- Repite lo anterior para insertar 300 nuevos nombres de personas.

5.- Al comprobar el contenido de la tabla **personas** observarás que se ha producido un error en los datos cargados. A todas las personas les hemos asignado el sexo 'M'.

Modifica el procedimiento para que al cargar los nombres cargue correctamente el sexo (H si es un hombre y M si es una mujer).

6.- Ahora vamos a crear un nuevo procedimiento para poner apellidos a todas las personas con números comprendidos entre el número mínimo y el máximo que se pasen como parámetros (no vamos a poner nada para comprobar posibles errores). No lo vamos a hacer en el editor de procedimientos sino con el cliente **mysql** o con el editor de instrucciones SQL en Workbench.

```
delimiter //  
CREATE PROCEDURE carga_apellidos(IN inicio INT, IN fin INT)  
BEGIN  
  declare n int;  
  declare n2 int;  
  declare ape1 varchar(25);  
  DECLARE ape2 VARCHAR(25);  
  set n=inicio;  
  while (n <=fin) do  
    set n2=floor(rand()*65428);  
    set ape1=(select apellido from apellidos where n2>=desde AND n2<hasta);  
    set n2=floor(rand()*65428);  
    set ape2=(select apellido from apellidos where n2>=desde AND n2<hasta);  
    update personas set apellido1=ape1,apellido2=ape2 where num=n;  
    set n=n+1;  
  end while;  
END//
```

7.- Envía al servidor el código del procedimiento.

8.- Ejecuta el procedimiento para que se carguen los apellidos de las 100 primeras personas.

9.- Modifica el procedimiento para que los números aleatorios que se generan, lo hagan con respecto al valor máximo que hay en la columna **hasta**.

10.- Envía nuevamente el procedimiento al servidor y vuelve a ejecutar para que se generen apellidos para las personas con números entre 50 y 300.