**Se trata de:**

* Realizar modificaciones de datos en tablas usando la instrucción UPDATE

**Objetivos:**

* Editar y ejecutar instrucciones UPDATE sobre una tabla de acuerdo a los requisitos exigidos.
* Modificar valores de filas de una tabla usando SET y especificando valores constantes.
* Modificar valores de columnas una fila a partir de expresiones que utilizan expresiones con valores de columnas y/o valores devueltos por funciones.

**Ejercicios**

Esta actividad se realizará sobre la base de datos **concursomusica.**

Debes saber escribir las instrucciones SQL para realizar cada una de las operaciones siguientes.

1.-Establece que el nombre y apellidos del usuario cuyo identificador de usuario es **21meko** sea **Lourdes Miera Campillo.**

update usuarios set nombre="Lourdes" and apellidos="Miera Campillo" where user="21meko"

2.- Comprueba mediante una SELECT cuál es el identificador de usuario de nombre y apellidos **Lourdes Hernández González** y que **votos** ha dado esta usuaria.

select user, cancion, fecha from votos inner join usuarios on user=usuario where nombre="Lourdes" and apellidos="Hernández González"

Modifica el identificador de usuario de **Lourdes Hernández González** para que sea **13LourHerna.** En la instrucción debes usar el nombre y apellidos del usuario en la condición que filtra el usuario en el que se va a modificar su **user**.

update usuarios set user="13LourHerna" where nombre="Lourdes" and apellidos="Hernandez Gonzalez";

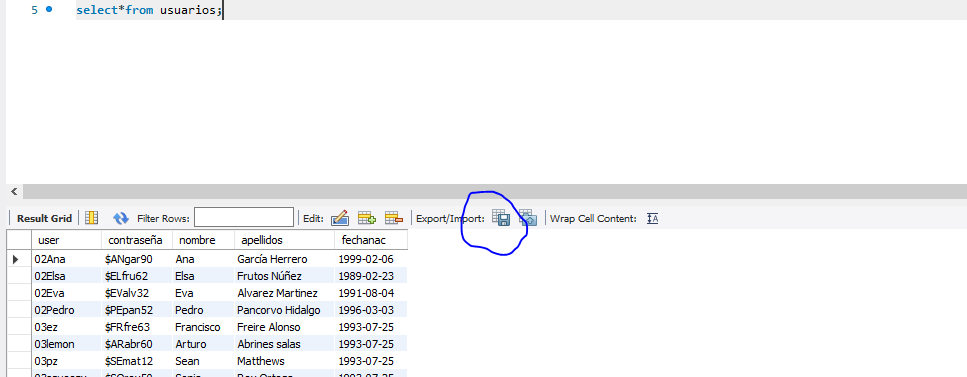
Comprueba cómo se han modificado los votos registrados de ese usuario con el nuevo identificador de usuario.

3.- Modifica las contraseñas de todos los usuarios para que estén formadas por la concatenación de **$**, seguido de los dos primeros caracteres del nombre en mayúsculas, los tres primeros de los apellidos en minúsculas y un número aleatorio entre **10 y 99**.

La formula para generar numeros aleatorios entre m y n es: rand()\*(n-m+1)+m

update usuarios set contraseña=concat("$",left(ucase(nombre),2),left(lcase(apellidos),3),round(rand()\*(99-10+1)+10));

Guarda en un archivo **usuarios.csv** los datos de todos los usuarios. Copia este archivo en cualquier carpeta para que sepas la contraseña sin encriptar que se vaya a usar en algún ejercicio de próximas actividades.



4.- Realiza una consulta del resultado de cifrar con MD5 las contraseñas de todos los usuarios.

update usuarios set contraseña=md5(contraseña);

Modifica las contraseñas de todos los usuarios para que contengan las contraseñas actuales encriptadas con MD5. Antes de realizarlo, debes modificar el tipo de la columna **contraseña** de la tabla **usuarios** para que pueda almacenar 32 caracteres.

5.- Antes de realizar lo siguiente, obtén el código del grupo **Repion.**

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Modifica la canción titulada ‘**La Divina Señal’** del grupo **Repion** por la canción **‘La Bahía’** del mismo grupo. La duración de esta canción será **5 minutos y 25 segundos**. Hay que comprobar que la canción que se modifica se titula **La Divina Señal** y que el código del grupo es el del **Repion** obtenido anteriormente**.** Aunque podría funcionar sin comprobar el código del grupo, no se debe hacer así por si hubiera títulos de canciones repetidos y de varios grupos.

update canciones set titulo="La Bahía",duracion="00:05:25" where titulo="La Divina Señal" and grupo=23;

6.- Establece que todas las canciones del grupo **Repion (usando el código de grupo anterior)** tengan (en la tabla canciones) tres votos más que los que tienen actualmente.

update canciones set total\_votos=total\_votos+3 where grupo=23;

7.- Modifica los votos dados antes de septiembre de 2022 para que tengan, dentro de la fecha del voto, una fecha una semana superior a la que tienen actualmente.

update votos set fecha=adddate(fecha, interval 1 week) where fecha<"2022-09-01";

Para asegurar la correcta realización del proceso, antes de proceder a modificar, haz una SELECT en la que obtengas la fecha que tienen estos votos y la fecha con la modificación.

*Al realizar esto podría darse el caso de que se produzca algún DUPLICATE KEY y no se ejecute UPDATE. Podría ocurrir que, al modificar la fecha del voto de un usuario, la nueva fecha coincidiera con la fecha de otros votos del mismo usuario y eso sería rechazado por las condiciones de* ***primary key*** *de la tabla.*

8.- Modifica los 10 votos de fecha más reciente, para que tengan una fecha dos semanas superior a la que tienen actualmente.

update votos set fecha=adddate(fecha, interval 2 week) order by fecha desc limit 10;

9.- Tras la modificación anterior, te debería quedar algún voto que corresponde a una fecha superior a la actual. Modifica todos los votos que tengan una fecha superior a la actual con la fecha actual.

10.- Obtén con una SELECT los datos de los votos dados el 3 y el 4 de octubre.

Modifica esos votos para que contengan en la fecha un día más que el registrado.

Al hacer esto, debería darte un error ya que al intentar modificar la fecha del voto dado por **17Fabri** el 3 de octubre, se modificaría esa fecha a 4 de octubre y se produciría un DUPLICATE KEY ya que en esa fecha **17Fabri** ya tiene un voto.

Realiza la modificación de forma que se empiece modificando primero las fechas mayores y termine con las fechas menores.