

Hoja de ejercicios 1-03

JDBC-Consultas en bases de datos

Objetivos:

- Preparar una sentencia SQL
- Preparar una sentencia SQL parametrizada
- Cargar valores en un sentencia SQL parametrizada
- Enviar sentencias SQL al SGBD
- Recibir resultados de sentencias SQL
- Procesar los resultados de una sentencia SQL
- Controlar excepciones de conexión.

RECURSOS

En los ejercicios de esta actividad tienes que usar la base de datos MySQL **concursumusica** disponible en el servidor MySQL local. Si no tienes la base de datos **concursumusica** debes importarla mediante el archivo **concursumusicaMySQL.sql**.

En esta base de datos se gestiona un concurso de música que se realiza en una página web. En esa página web los usuarios registrados pueden votar canciones de grupos de música (las canciones registradas en la base de datos). De cada grupo puede haber varias canciones.

La tabla **grupos** contiene información sobre todos los grupos y artistas en solitario que participan con alguna canción en el concurso (pueden participar con varias). Cuando se trata de un artista en solitario, indica con false o 0 en la columna **esgrupo**. Por defecto en esta columna se carga true o 1. En **fechaEstreno** se registra la fecha de la primera actuación en directo del grupo y en **annoGrab** el año en que grabaron el primer disco. El identificador del grupo **codgrupo** es una columna numérica y autoincrementada.

De cada grupo, se tiene en la tabla **componentes** información sobre los componentes del grupo. En la columna **función** se tienen valores como **batería, vocalista, voz y guitarra, etc.**

En la tabla **canciones** se tienen todas las canciones que se pueden votar. El identificador de cada canción es el número de canción (**numcancion que es autoincrementada**). Hay una columna total de votos que actualmente está a cero pero que debería contener la contabilización de todos los votos que se han dado a cada canción. La duración es una columna de tipo **TIME**, por tanto, se representan sus datos en formato **HH:MM:SS**. La columna grupo es clave ajena y contiene el identificador del grupo al que pertenece la canción.

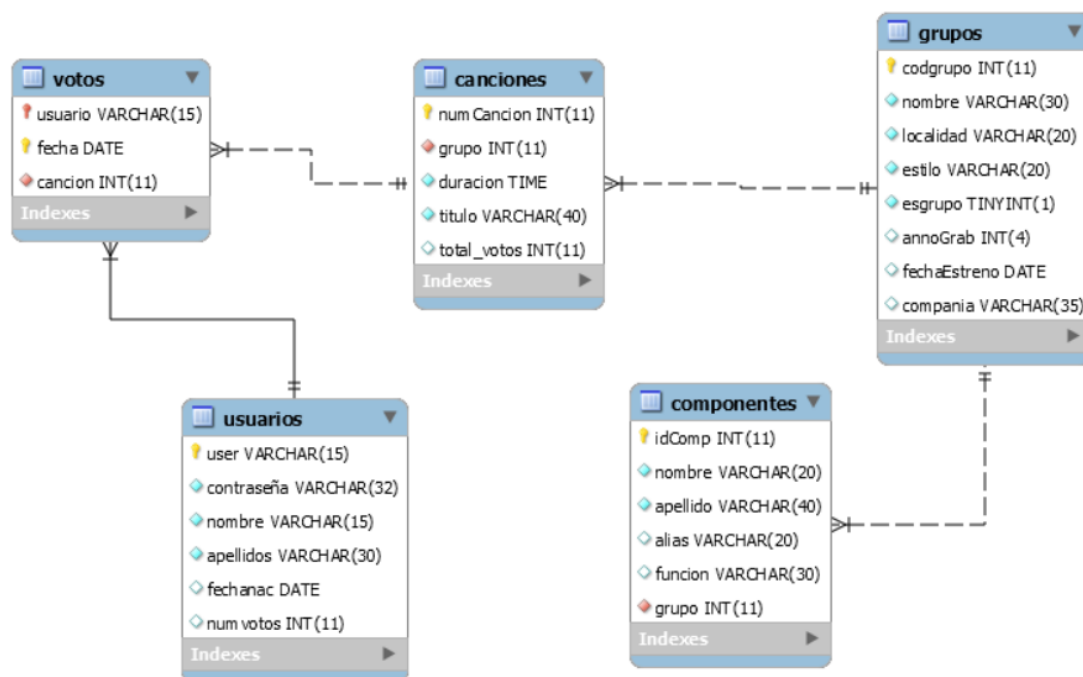
Los usuarios de la web se almacenan en la tabla **usuarios**. Cada usuario elige su identificador (**user**) y su contraseña. La contraseña debe almacenarse con encriptación

Hoja de ejercicios 1-03

MD5. En la tabla, la columna **numero_votos** debe contener el número total de votos que ha dado un usuario.

En la tabla **votos** se inserta una fila cada vez que un usuario vota una canción. En esta tabla es **PK fecha+usuario**, por lo que un usuario no pueda dar más de un voto en un mismo día. En la columna **cancion** se registra el número o identificador de canción votada en cada voto.

El esquema relacional correspondiente a esta base de datos es:



EJERCICIO 1

Realiza un programa que permite añadir un usuario a la tabla **usuarios** con los datos recogidos por teclado.

No es necesario validar los datos introducidos.

Si que hay que comprobar si al hacer la inserción se produce excepción por clave duplicada debido a que el user ya exista. El código de error MySQL por clave duplicada es 1062. El método de **SQLException** que devuelve el código de error que ha generado la excepción es **getErrorCode()**.

Hoja de ejercicios 1-03

EJERCICIO 2

En el programa a realizar, cada método de acceso a BD debe iniciar conexión y cerrarla al finalizar su ejecución.

Realiza un programa que presenta el siguiente menú:

- 1.- Listado de grupos
- 2.- Listado de canciones
- 3.- Número de canciones por grupo
- 4.- Canciones de un grupo
- 5.- Las 5 canciones más votadas
- 6.- Grupos sin canciones
- 7.- Los últimos 5 votos
- 8.- Eliminar canciones de un grupo
- 9.- Modificar datos de grupo

Y escribe en pantalla los resultados de la opción seleccionada.

Opción 1: El listado de grupos mostrará el código del grupo, el nombre, la localidad y el estilo y debe estar ordenado por el código.

Para esta opción como para las siguientes se inicia la conexión y se cierra al iniciar y terminar el proceso correspondiente a la opción.

Opción 2: El listado de canciones debe estar ordenado por nombre de grupo y debe tener este formato:

```
Astrud
    Bailando
    Todo nos parece una mierda
Carlos Sadness
    Qué electricidad
    Perseide
    . . . . .
```

Opción 3: Deben mostrarse todos los grupos y, para cada grupo, el número de canciones que tiene.

Opción 4: Se pide un nombre de grupo y muestran los datos de las canciones del grupo (número, título y duración).

Opción 5: Se deben mostrar, para las canciones con más votos, títulos de canciones y nombres de los grupos.

Opción 6: Se muestra un listado de nombres de grupos sin canciones en la tabla canciones

Hoja de ejercicios 1-03

Opción 7: Se muestran datos de los 5 votos más recientes. Hay que mostrar el título de la canción votada y el grupo al que pertenece.

Opción 8: Se pide por teclado el nombre de un grupo y, si existe, se eliminan todas las canciones del grupo. Como está restringido el borrado de canciones si tienen votos, se han de eliminar primero los votos de las canciones del grupo.

Opción 9: Se pide por teclado el nombre de un grupo y, si existe, se escribe en pantalla, el estilo, el año de primera grabación de un disco, la fecha de la primera actuación, la localidad y la compañía. A continuación, se escribe un menú preguntando cuál de los anteriores datos se quiere modificar, incluido el nombre del grupo, y se modifica ese dato con el nuevo valor que se introduzca por teclado.

EJERCICIO 3

Realiza un programa que pide las credenciales de un usuario y, si son correctas:

- Permite que el usuario vote, en la fecha actual, la canción que tenga el número de canción que indique por teclado (añade un voto en la tabla **votos**).
- Contabiliza el voto del usuario, es decir, incrementa en uno el valor existente en la columna **num_votos** de la tabla **usuarios** para el usuario que ha realizado el voto.
- Contabiliza el voto de la canción.
- Muestra el título de la canción votada, el grupo al que pertenece y el puesto en que está en la clasificación de votos.