

## Actividad de aprendizaje 4-03

### Se trata de:

- Realizar consultas SELECT sobre una sola tabla con condiciones de selección simples.

### Objetivos:

- Editar y ejecutar instrucciones SELECT sobre una tabla de acuerdo a los requisitos exigidos.
- Utilizar la cláusula WHERE dentro de la instrucción SELECT.
- Seleccionar filas por condiciones sobre columnas de tipo fecha y hora.
- Seleccionar filas por condiciones sobre columnas de tipo booleano.
- Seleccionar filas por condiciones sobre columnas de texto.
- Seleccionar filas por condiciones sobre columnas numéricas.

### Descripción y diagrama relacional de la base de datos concursomusica

**En esta actividad se va a usar por primera vez la base de datos concursomusica cuyo archivo de importación (.sql) tienes disponible como adjunto a la tarea.**

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** que es una columna numérica y autoincrementada.

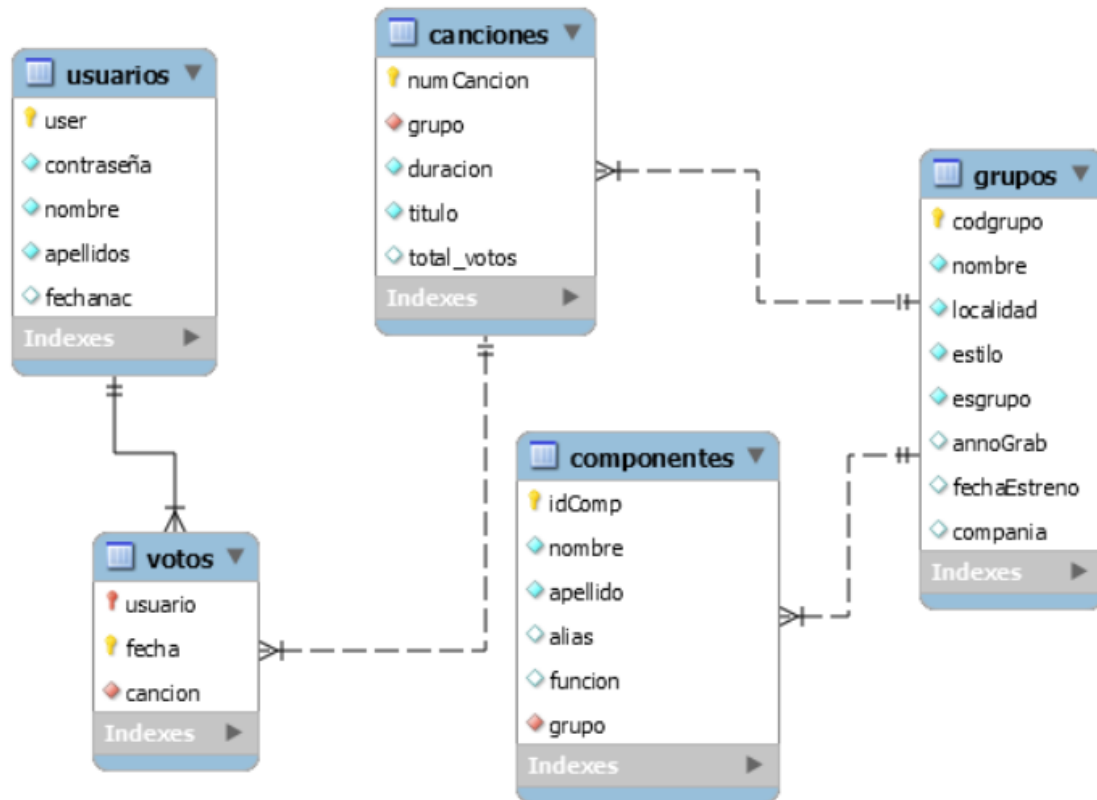
De cada grupo, se tiene en la tabla **componentes** información sobre los componentes del grupo. En la columna **función** se tienen valores como **batería, vocalista, voz y guitarra, etc.**

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. En futuras prácticas ya veremos como se actualiza esa información a partir de todos los votos registrados o, mucho mejor, cada vez que se añade un voto. 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, en un futuro, quedará almacenada con encriptación **MD5**.

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:



### Ejercicios

1.- Obtén de todas las canciones que duran más de 3 minutos: el número de canción, el identificador de grupo al que pertenecen, la duración y el título, ordenando por duración de mayor a menor.

```
select numCancion, grupo, duracion, titulo from canciones where duracion>='00:03:00' order by duracion desc
```

2.- Obtén los datos de todos los votos realizados antes del año 2018.

```
select * from votos where fecha<'2018-01-01'
```

3.- Obtén los datos de todos los votos realizados el día 2 de enero de 2018.

```
select * from votos where fecha='2018-01-02'
```

4.- Sabiendo que la fecha actual se obtiene en **MySQL** con la función **curdate()**, obtén los datos de todos los votos realizados en la fecha actual.

```
select * from votos where fecha=curdate();
```

5.- Obtén los datos de los votos que constan como realizados después de la fecha actual.

**select \* from votos where fecha>curdate();**

6.- Obtén los datos del voto que se realizó por primera vez (el primero por fecha).

**SELECT \* FROM votos ORDER BY fecha ASC LIMIT 1;**

7.- Obtén los nombres de todos los grupos que han grabado su primer disco antes de 2006.

**SELECT nombre FROM grupos WHERE annoGrab<2006;**

8.- Obtén los nombres de todos los grupos que han realizado su primera actuación en concierto antes de 2006.

**SELECT nombre FROM grupos WHERE fechaEstreno<'2006-01-01';**

9.- Obtén los nombres de todos los grupos de Madrid.

**SELECT nombre FROM grupos WHERE localidad="Madrid";**

10.- Obtén los nombres de todos los cantantes en solitario que hay en la tabla grupos.

**SELECT nombre FROM grupos WHERE esGrupo=false;**

11.- Obtén los nombres de todos los grupos cuyo nombre comienza por la letra V.

**SELECT nombre FROM grupos WHERE nombre LIKE 'V%';**

12.- Obtén los nombres y apellidos de todos los componentes del grupo con identificador 21.

**SELECT nombre, apellido FROM componentes WHERE grupo=21;**

13.- Obtén los nombres de todos los grupos cuyo estilo sea **pop** o mezcla de **pop** con otros estilos.

**SELECT nombre FROM grupos WHERE estilo LIKE '%pop%';**

14.- Obtén los datos de todas las canciones del grupo con identificador 21.

**SELECT \* FROM canciones WHERE grupo=21;**

15.- Obtén los datos de las cinco canciones de menor duración.

**SELECT \* FROM canciones ORDER BY duracion LIMIT 5;**