

Actividad de aprendizaje 3-03

Se trata de:

- Realizar el diseño físico de una base de datos para gestionar un concurso de música en el que las votaciones se producen por usuarios registrados en una web.

Objetivos:

- Interpretar el diseño lógico de una base de datos para implementar su diseño físico.
- Crear las tablas a partir del diseño lógico.
- Definir columnas y tipos de datos a partir del diseño lógico y de la descripción de funcionalidad.
- Establecer restricciones de columnas a partir del diseño lógico y de la descripción de funcionalidad.
- Comprobar que una base de datos se ha implementado correctamente.

Procedimiento

Lee primero la descripción de funcionalidad de la base de datos que recoge la información necesaria para que los usuarios puedan ver información sobre las canciones que se votan y sobre que grupos o intérpretes en solitario cantan o tocan cada canción.

En esta base de datos **music_contest** 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, se indica con false o 0 en la columna **esgrupo**. Por defecto en esta columna se carga true o 1, es decir, que corresponde a un grupo. 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.

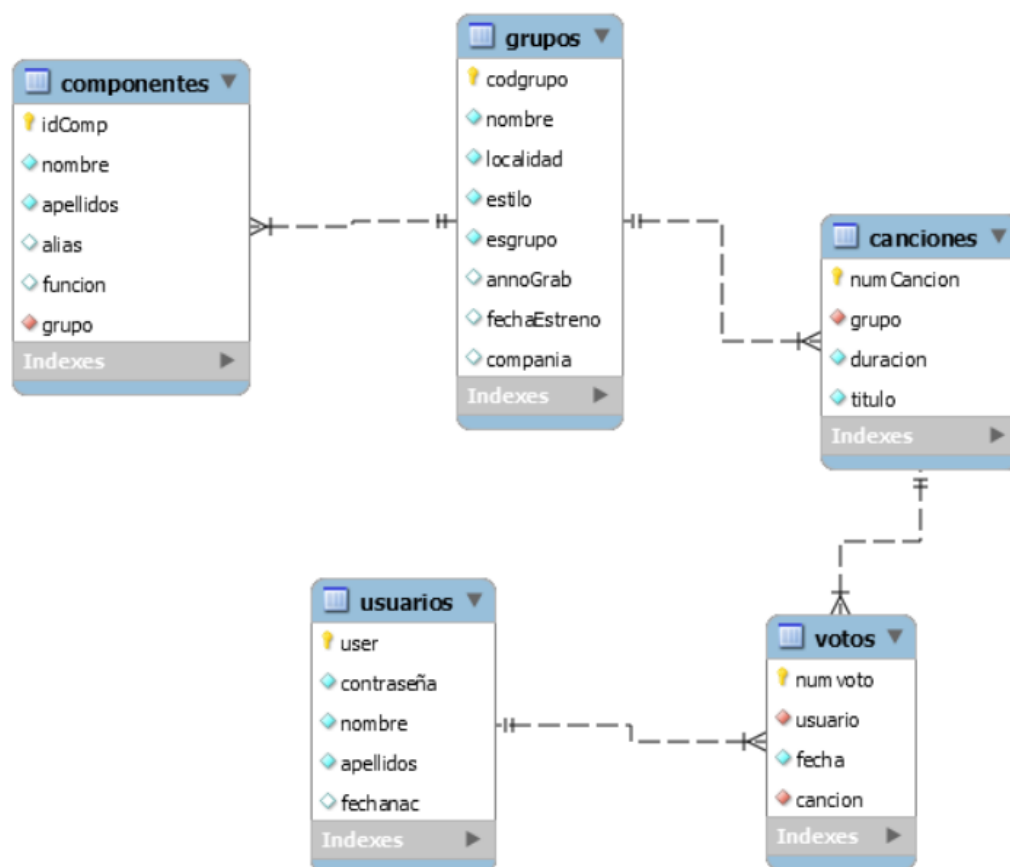
De cada grupo, se tiene en la tabla **componentes** información sobre los componentes del grupo. En la columna **función** se tienen valores como **batería, vocalista, voz y guitarra, etc.**

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**). La duración es una columna que debe permitir almacenar los minutos y segundos que dura una canción. La columna grupo contiene el identificador del grupo que interpreta la canción.

Los usuarios de la web se almacenan en la tabla **usuarios**. Cada usuario elige su identificador (**user**) y su contraseña (que deberá tener un mínimo de 8 caracteres y un máximo de 20). En un futuro, la contraseña deberá quedar almacenada con encriptación **MD5**. Se debería establecer un control para que una misma persona no se pueda registrar con varias identidades de usuario. Por ahora, se permite registrar a cualquiera.

En la tabla **votos** se insertará una fila cada vez que un usuario vota una canción. En esta tabla, un usuario no podrá dar más de un voto en un mismo día. Por ello, el conjunto fecha del voto y usuario no se podrán repetir. En la columna **cancion** se registra el número de canción votada en cada voto. La clave primaria debe ser una columna autoincrementada.

El esquema relacional correspondiente a esta base de datos es:



Escribe las instrucciones SQL necesarias para crear la base de datos correspondiente al grafo relacional y descripción de funcionalidad de la base de datos del concurso de música. Se deben:

- Elegir los tipos de columnas adecuados.
- Establecer las restricciones NOT NULL
- Establecer las restricciones DEFAULT o valor por defecto
- Establecer los índices normales que se indican, dándoles un nombre

- Establecer los índices UNIQUE dándoles un nombre
- Establecer las PRIMARY KEY
- Establecer las claves ajenas y reglas de integridad referencial. Dar un nombre a todas las relaciones de clave ajena.
- No se pueden usar cualificadores de tabla en las instrucciones
- No se puede usar comillas simples en los nombres de los objetos.

TABLA GRUPOS

```
CREATE TABLE grupos (  
codgrupo INT NOT NULL AUTO_INCREMENT,  
nombre VARCHAR(30) NOT NULL,  
localidad VARCHAR(20) NOT NULL,  
estilo VARCHAR(20) NOT NULL,  
esgrupo TINYINT NOT NULL DEFAULT '1',  
annoGrab INT(4),  
fechaEstreno DATE,  
compania VARCHAR(35),  
PRIMARY KEY (codgrupo));
```

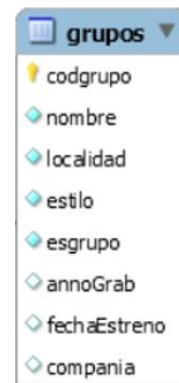


TABLA USUARIOS

```
CREATE TABLE usuarios (  
user VARCHAR(15) NOT NULL,  
contrasena VARCHAR(20) NOT NULL,  
nombre VARCHAR(15) NOT NULL,  
apellidos VARCHAR(30) NOT NULL,  
fechanac DATE,  
PRIMARY KEY (user),  
UNIQUE(nombre,apellidos)  
);
```

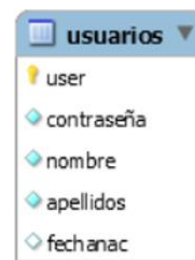
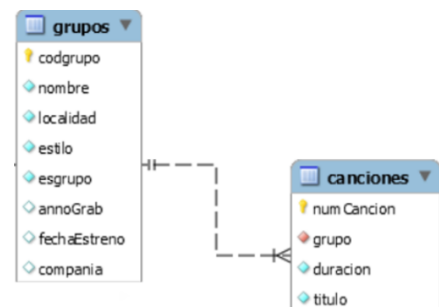


TABLA CANCIONES

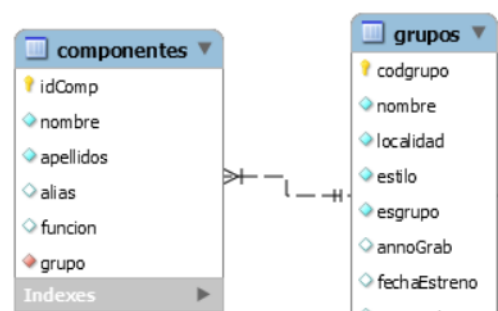
```
CREATE TABLE canciones (  
numCancion INT NOT NULL AUTO_INCREMENT,  
grupo INT NOT NULL,  
duracion TIME NOT NULL,  
titulo VARCHAR(40) NOT NULL,  
PRIMARY KEY (numCancion),
```



```
CONSTRAINT fk_cancion_grupo FOREIGN KEY (grupo) REFERENCES  
grupos (codgrupo) );
```

TABLA COMPONENTES

```
CREATE TABLE componentes (  
idComp INT NOT NULL AUTO_INCREMENT,
```



```
nombre VARCHAR(20) NOT NULL,  
apellidos VARCHAR(40) NOT NULL,  
alias VARCHAR(20),  
función VARCHAR(30),  
grupo INT NOT NULL,  
PRIMARY KEY (idComp),  
CONSTRAINT fk_componente_grupo FOREIGN KEY (grupo) REFERENCES  
grupos (codgrupo) );
```

TABLA VOTOS

```
CREATE TABLE votos (  
numvoto INT NOT NULL AUTO_INCREMENT,  
usuario VARCHAR(15) NOT NULL,  
fecha DATE NOT NULL,  
cancion INT NOT NULL,  
PRIMARY KEY (numvoto),  
UNIQUE(usuario,fecha),  
CONSTRAINT fk_voto_canciones FOREIGN KEY (cancion) REFERENCES  
canciones (numCancion),  
CONSTRAINT fk_voto_usuarios FOREIGN KEY (usuario) REFERENCES  
usuarios (user)  
);
```

