Estructura de la base de datos

Tenemos 16 tablas: **users, documents, comments, chats, user\_chat, messages, friend\_request, friends, offers, user\_offer, videos, visualized\_videos, subject, quizzzes, answers, user\_answer.**

**La tabla users**, es la contiene toda la información sobre los usuarios y consta de 9 campos. El primer campo **id** es la clave primaria y es autoincrement ya que se le asigna automáticamente a cada usuario que se crea una cuenta para que así no haya dos que tengan el mismo, el segundo campo **name** hace referencia al nombre de cada usuario, ese decir, el nombre que van a ver el resto de usuarios, el tercer campo **password**, es la contraseña de la cuenta de cada usuario, la cual ira cifrada en la base de datos, el cuarto campo **role** es el rol que tiene cada usuario(standard si es un usuario normal y corriente, admin si es un administrador que tiene control total de todos los campos o company si el usuario es una empresa, ya que así son estos los únicos que pueden subir documentos relacionados con prácticas de empresa u ofertas), el quinto campo **profile\_pic**, almacena el path en el que se encuentra la foto de perfil de cada usuario, el sexto campo  **created\_at** hace referencia a la fecha en la que se creó la cuenta de cada usuario, el séptimo campo **updated\_at** almacena la fecha en al que se ha modificado la cuenta del usuario, y por último el octavo campo **ban** hace referencia a si el usuario esta baneado o no(0 por defecto que significa que no está baneado y 1 refiriéndose a que sí está baneado).

**La tabla videos**, almacena toda la información de los videos subidos y consta de 8 campos. El primer campo **id** es la clave primaria y es autoincrement ya que se le asigna a cada video nada más se suba a nuestra aplicación y no queremos que haya 2 con el mismo valor, además, así es más fácil la comunicación entre tablas. El segundo capo **owner\_id** almacena el id del usuario que ha subido el video y actúa como índice ya que así podemos comunicarnos mejor con las otras tablas a la hora de buscarlo, el tercer campo **subject\_id** almacena el tema del video, ya que nos interesa que los videos con temas iguales/parecidos se visualicen en los mismos sitios, también actúa como índice, el cuarto campo **tittle** guarda el nombre del vídeo(el que le decide dar el usuario al subirlo), el quinto campo **description** es donde se guarda la descripción que le da el usuario al video al subirlo, el sexto campo **file\_path** almacena el path donde se encuentra el vídeo, el séptimo campo **created\_at** guarda la fecha en la que se subió el vídeo, y por último el octavo campo **status** contiene el estado del vídeo(valid si es válido, pending si está pendiente de revisión por x motivos y bloqued si está bloqueado).

**La tabla subject** se encarga de almacenar el tema al que corresponde el video y consta de 2 campos. Un primer campo **id** que actúa como clave primaria y un segundo campo **name** que actúa como nombre del tema (ya que los videos van a estar distribuidos según temas).

**La tabla visualized\_videos** es una tabla que utilizamos de forma auxiliar para que nos sea fácil relacionar los videos que ha visto cada usuario específico, ya que almacena en un primer campo **user­\_id** el id del usuario siendo este un índice y en su segundo campo **video\_id** el id del video el cual es también un índice y un tercer campo **fav** que nos indica si el usuario le ha dado o no a favorito (en tal caso se le mostraría en un apartado de vídeos favoritos). La clave primaria son los 2 primeros campos campos.

**Cuestionarios:**

**Hemos pensado que en cada vídeo pueda haber varias preguntas tipo test, estas preguntas se almacenan en la tabla quizzes con un id único y con un campo que almacena el momento del vídeo en el que debe aparecer la pregunta. Las respuestas se almacenan en otra tabla y guardan el id de la pregunta (quiz) al que pertenecen. La pregunta almacenada en la tabla quizzes es la que conoce el id de la respuesta correcta. La respuesta que da el usuario se guarda en la tabla user\_answer que analizaremos luego.**

**La tabla quizzes** almacena la información relacionada con los cuestionarios que se encuentran en los vídeos y consta de 5 campos. El primer campo **id** es la clave primaria, el segundo campo **video­\_id** almacena el id del video en el cual se encuentra el cuestionario, actúa también como índice, el tercer campo **correct­­\_answer\_­id** que almacena el id de la respuesta correcta a cada pregunta del cuestionario y actúa como índice, el cuarto campo **question\_text** que guarda el texto de la pregunta y por último el campo **video\_stamp** que almacena el momento del video en el que sale.

**La tabla answers** se encarga de almacenar las respuestas de los cuestionarios y consta de 3 campos. El primer campo **id** actúa como clave primaria, el segundo campo **quiz\_id** que contiene el id del cuestionario al que pertenece la respuesta y actúa como índice, y un tercer campo **answer\_text** que guarda el texto de la respuesta en cuestión.

**La tabla user\_answer** guarda las respuestas que ha dado el usuario y consta de 2 campos. El primer campo **user\_id** que contiene el id del usuario y actúa como índice, y el segundo campo **answer\_id** que almacena el id de la respuesta y actúa como índice. Ambos campos componen la clave primaria.

**La tabla comments** es la encargada de gestionar los comentarios de los videos y consta de 5 campos. El primer campo **id** funciona como clave primaria y se le asigna automáticamente se crea el comentario ya que no queremos que haya dos distintos con la misma id, el segundo campo **user\_id**  almacena el id del usuario que ha escrito ese comentario y actúa como índice, el tercer campo **vide­o­\_id** es el id del video correspondiente en el que se ha escrito ese comentario y también actúa como índice, el cuarto campo **comment\_text** hace referencia al texto que puede escribir el usuario en el comentario y por último, el quinto campo **posted\_at** almacena la fecha en la que se publicó/escribió el comentario.

**La tabla chats** es la que se encarga de los chats entre varios usuarios y consta de 4 campos. El primer campo **id** que identifica cada chat único y actúa como clave primaria, un segundo campo **created\_at** que almacena la fecha de creación del chat, un tercer campo **name** que sirve para dar nombre al chat, un cuarto y último campo **description** que almacena una descripción del chat.

**La tabla user\_chat** almacena los chats que tiene cada usuario y consta de 2 campos. El primer campo **user\_id** que guarda el id del usuario y actúa como índice y un segundo y último campo **chat\_id** que guarda el id del chat. La clave primaria son ambos campos juntos.

**La tabla messages** se encarga de almacenar la información sobre los mensajes según el chat y consta de 4 campos. El primer campo **id** que actúa como clave primaria, un segundo campo **chat\_id** que almacena el id del chat al cual corresponde el mensaje y actúa como índice, el tercer campo **msg\_text** guarda el texto del mensaje y un cuarto campo **sent\_at** que almacena la fecha en la que se envió el mensaje.

**La tabla friend\_request** guarda toda la información relacionada con la petición de amistad entre dos usuarios y consta de 5 campos. El primer campo **status** guarda el estado de la petición (accepted si el usuario que la recibe la ha aceptado, pending si el usuario que la recibe todavía no la ha aceptado y bloqued si un usuario decide bloquear a otro), el segundo campo **user\_to** guarda el id del usuario al cual se ha enviado la petición de amistad por ello actúa también como índice, el tercer campo **user\_from** guarda el id del usuario que ha enviado la petición de amistad y actúa también como índice, el cuarto campo **sent\_at** almacena la fecha en la que se envió la petición de amistad y el quinto y último campo **updated\_at** guarda la fecha en la que el destinatario ha aceptado/rechazado la petición. La clave primaria es la conjunción de los campos user\_to y user\_from.

**La tabla friends**  contiene los ids de los usuarios que son amigos (entre ellos) y tiene 2 campos. El primero **user1\_id** que guarda el id de un usuario y actúa como índice y el segundo **user2\_id** que guarda el id del otro usuario y también como índice. Ambos campos juntos actúan como clave primaria.

**La tabla user\_offer** almacena por cada usuario las ofertas que tiene marcadas y tiene 2 campos. El primer campo **user\_id** que almacena el id del usuario y actúa como índice y el segundo campo **offer\_id** que guarda el id de la oferta que ha marcado el usuario y actúa también como índice. Esta tabla nos ayuda a poder sacar fácilmente todas las ofertas seleccionadas por un mismo usuario. La clave primaria son los dos campos juntos.

**La tabla offers** tiene toda la información sobre las ofertas y tiene 11 campos. El primer campo **id** actúa como clave primaria a que identifica a cada oferta como única, el segundo campo **owner\_id** guarda el id de la persona (en nuestro caso la del rol de company) que ha subido la oferta y actúa como índice, el tercer campo **offer\_type** que guarda qué tipo de oferta es (job si es de trabajo o internship si es de prácticas), el cuarto campo **description** guarda la descripción que se le puede poner a cada oferta (la pone la persona que lo suba), el quinto campo **created\_at** almacena la fecha en la que se subió la oferta a la app, el sexto campo  **updated\_at** almacena la fecha en la que se modificó/actualizó la oferta (también lo hace el que lo sube o el admin si se ha visto obligado a modificarla), el séptimo campo **name** contiene el nombre de la oferta, el octavo campo **start\_at** guarda la fecha en la que inician las prácticas/ofertas (por si quieres filtrar según tu disponibilidad), el noveno campo **end\_at** guarda cuando acabarían las practicas/trabajo (por si quieres filtrar), el décimo campo **paid** indica, en caso de ser prácticas si son remuneradas o no y el decimoprimer campo y último **vacants**  que almacena la cantidad plazas libres que tiene la empresa para contratar.