

# Administración de página de galería de imágenes

**Preentrega Proyecto – SQL**

**Alumno: Joaquín Sosa**

## **Introducción**

Este proyecto se centra en la base de datos usada para una página web cuyo propósito es funcionar como una galería de imágenes donde el usuario puede subir sus propias imágenes, buscar imágenes por tags, comentar en las imágenes y votar con un like o un dislike.

## **Objetivo**

El objetivo de este proyecto es hacer una base de datos eficiente para el manejo de los datos de esta página web, haciendo consultas optimizadas, utilizando mejores prácticas y dando una mejor experiencia al usuario.

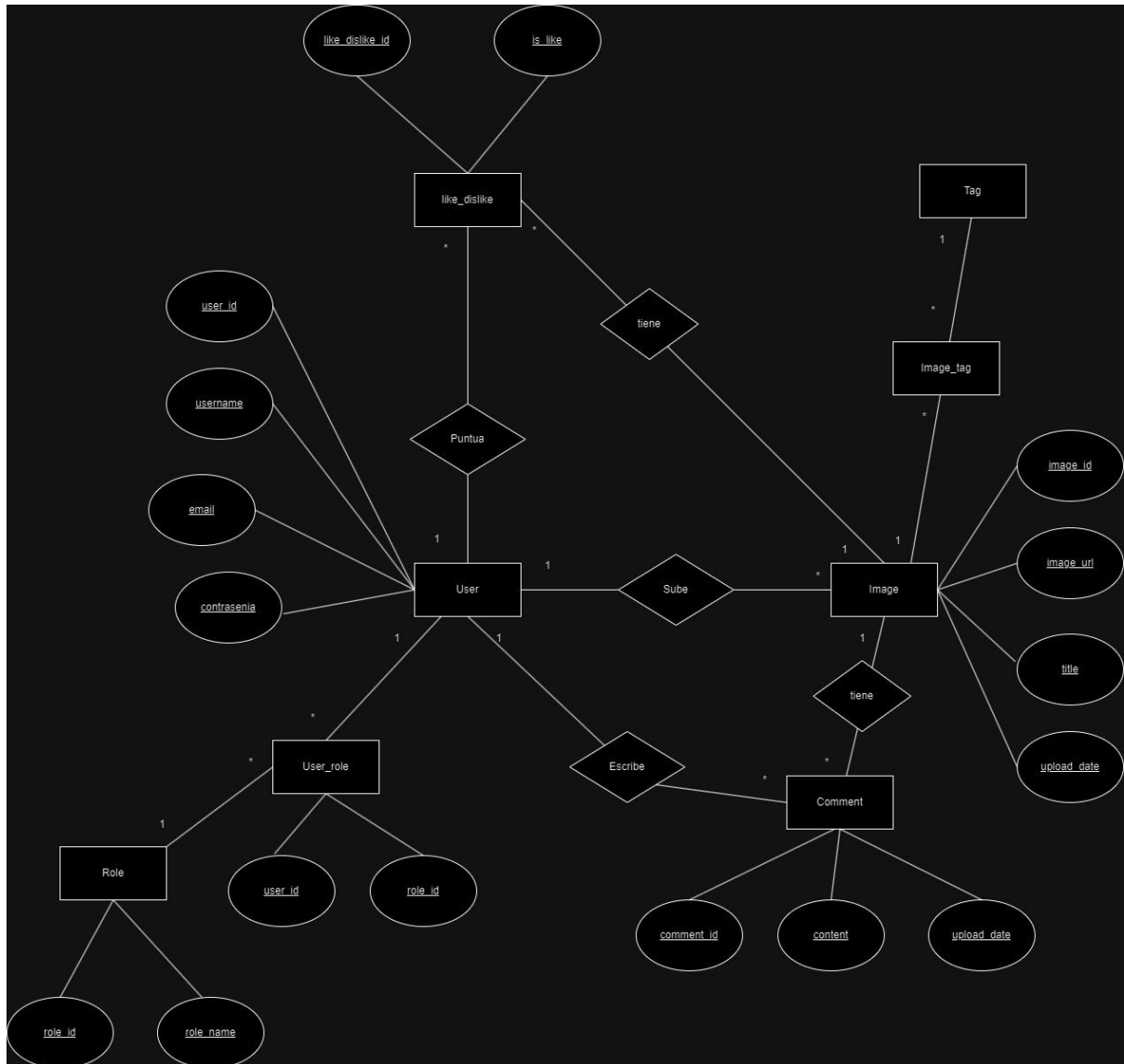
## **Situación Problemática**

Hace falta una estructura donde almacenar la información y una base de datos estructurada es ideal para almacenar los datos que se usaran en esta página web, no solo eso si no que también hace que sea fácil de organizar y buscar la información que se necesite haciendo de esta una muy buena solución a como guardar y acceder a la información de los usuarios.

## **Modelo de Negocio**

La organización que usa esta solución es una empresa cuyo proyecto es crear esta plataforma de contenido visual donde la gente pueda interactuar con otros, subir imágenes, comentar y votar.

## Diagrama Entidad-Relación



## Tablas

### Tabla

#### User

Contiene información de cada usuario

PK	COLUMN	TYPE	NOT NULL	UNIQUE	LEN	NOTES
TRUE	User_id	INT	TRUE	TRUE		ID DEL USUARIO
	username	VARCHAR	TRUE	TRUE	50	NOMBRE
	email	VARCHAR	TRUE	TRUE	50	CORREO ELECTRÓNICO DEL USUARIO
	contrasenia	VARCHAR	TRUE		50	CONTRASEÑA DEL USUARIO

### Tabla

#### Role

CONTIENE LOS ROLES QUE PUEDE TENER UN USUARIO

PK	COLUMN	TYPE	NOT NULL	UNIQUE	LEN	NOTES
TRUE	role_id	INT	TRUE	TRUE		ID DEL PRESUPUESTO
	role_name	VARCHAR	TRUE	TRUE		NOMBRE DEL ROL

Tabla  
User\_role

TABLA INTERMEDIA PARA  
REPERESNTAR LA RELACIÓN MUCHOS A  
MUCHOS ENTRE USER Y ROLE

PK	COLUMN	TYPE	NOT NULL	UNI QUE	LE N	NOTES
FK	user_id	INT				ID DEL USUARIO
FK	role_id	INT				ID DEL ROL

Tabla  
Image

CONTIENE LA  
INFORMACIÓN DETALLADA  
DE LAS IMAGENES QUE SE  
SUBEN A LA PÁGINA

PK	COLUMN	TYPE	NOT NULL	UNIQUE	LEN	NOTES
TRUE	Image_id	INT	TRUE	TRUE		ID DE LA IMAGEN
FK	user_id	INT				ID DEL USUARIO
	image_url	VARCHAR	TRUE		225	URL PATH DE LA IMAGEN
	title	VARCHAR			100	TITULO DE LA IMAGEN
	upload_date	DATE				FECHA EN LA QUE SE SUBIO LA IMAGEN

## Tabla Comment

CONTIENE LA  
INFORMACIÓN  
DETALLADA DE LOS  
COMENTARIOS QUE  
HAY EN LAS  
IMAGENES

PK	COLUMN	TYPE	NOT NULL	UNIQUE	LEN	NOTES
TRUE	comment_id	INT	TRUE	TRUE		ID DEL COMENTARIO
FK	user_id	INT				ID DEL USUARIO
FK	image_id	INT				ID DE LA IMAGEN
	content	VARCHAR	TRUE		225	CONTENIDO DEL COMENTARIO
	upload_date	DATE				FECHA EN LA QUE SE SUBIO LA IMAGEN

## Tabla Tag

CONTIENE LA  
INFORMACIÓN DE LOS  
TAGS

PK	COLUMN	TYPE	NOT NULL	UNIQUE	LEN	NOTES
TRUE	tag_id	INT	TRUE	TRUE		ID DEL TAG
	tag_name	VARCHAR	TRUE	TRUE	255	NOMBRE DEL TAG

## Tabla

### Image\_tag

TABLA INTERMEDIA PARA  
REPERESNTAR LA RELACIÓN  
MUCHOS A MUCHOS ENTRE IMAGE Y  
TAG

PK	COLUMN	TYPE	NOT NULL	UNIQUE	LEN	NOTES
FK	tag_id	INT				ID DEL TAG
FK	image_id	INT				ID DE LA IMAGEN

## Tabla

### Like\_dislike

CONTIENE LA  
INFORMACIÓN DE LOS  
LIKES Y DISLIKES QUE  
HAY EN LAS IMAGENES

PK	COLUMN	TYPE	NOT NULL	UNIQUE	LEN	NOTES
TRUE	like_dislike_id	INT	TRUE	TRUE		ID DEL LIKE O DISLIKE
	is_like	BOOL				BOOLEANO QUE DETERMINA SI ES UN LIKE O NO
FK	image_id	INT				ID DE LA IMAGEN
FK	user_id	INT				ID DEL USUARIO

## **Vistas**

### 1- Imagenes\_con\_usuario\_vista

Objetivo: Mostrar todas las imágenes subidas a la base de datos junto con el usuario que las subió.

Tablas involucradas: Image y User.

### 2- Comentarios\_con\_usuario\_y\_imagen

Objetivo: Mostrar todos los comentarios, quien los escribió y en que imagen.

Tablas involucradas: Comment, User y Image.

### 3- imagenes\_con\_likes\_y\_dislikes

Objetivo: Mostrar todas las imágenes y la cantidad de likes y dislikes que tienen.

Tablas involucradas: Image y Like\_dislike.

### 4- usuarios\_con\_roles

Objetivo: Mostrar a todos los usuarios con sus roles.

Tablas involucradas: User y User\_role.

### 5- imagenes\_con\_tags

Objetivo: Mostrar todas las imagenes con sus tags

Tablas involucradas: Image, Image\_tag, Tag

## **Funciones**

### 1- contar\_comentarios\_por\_imagen\_fn(imageID INT)

Objetivo: Retorna el número total de comentarios asociados a una imagen específica.

Parámetro: Id de la imagen.

Tablas involucradas: Comment

2- usuario\_interactuo\_fn(userID INT, imageID INT)

Objetivo: Saber si un usuario específico interactuó con una imagen en particular (con like o dislike).

Parámetros: Id del usuario e id de la imagen.

Tablas involucradas: Like\_dislike

3- cantidad\_de\_likes\_de\_usuario\_fn(userID INT)

Objetivo: Saber cuántos likes dio un usuario específico en total.

Parámetro: Id del usuario.

Tablas involucradas: Like\_dislike

## **Stored Procedures**

1- insertar\_imagen\_sp

Objetivo: Insertar una imagen en la tabla Image.

Parámetros: título de la imagen, URL, descripción y fecha de creación.

Tablas involucradas: Image.

2- insertar\_comentario\_sp

Objetivo: Insertar un comentario en la tabla Comment.

Parámetros: Id del usuario, id de la imagen, contenido y fecha de creación.

Tablas involucradas: Comment.

## **Triggers**

1- evitar\_comentario\_vacio\_tr



Objetivo: Cuando se inserta un comentario en la tabla Comment este trigger se asegura de que no sea vacío, si es vacío manda un mensaje de error y se detiene el insert.

Tablas involucradas: Comment

2- actualizar\_total\_images\_tr

Objetivo: Cuando un usuario sube una imagen, se actualiza la columna total\_images en la tabla User.

Tablas involucradas: Image