

```
graph TD
    USERS[USERS]
    BIRDS[BIRDS]
    POSTS[POSTS]
    TAGS[TAGS]

    USERS -- "Likes (likeType)" --> POSTS
    USERS -- "Mentions (activeMention)" --> POSTS
    USERS -- "Does" --> POSTS
    USERS -- "Views (browser, device, viewDate, IP)" --> POSTS
    USERS -- "Preferences (activePreference)" --> BIRDS
    BIRDS -- "birdType" --> BIRDS
    POSTS -- "Tags" --> TAGS

    USERS --- activeUser
    USERS --- fullName
    USERS --- email
    USERS --- city
    BIRDS --- birdType
    POSTS --- title
    POSTS --- postText
    POSTS --- urlPhoto
    POSTS --- activePost
    TAGS --- activeTag
```

```
Users(idUser, fullName, email, city, activeUser),
Posts(isPost, title, postText, urlPhoto, idUser, activePost),
Birds(idBird, birdType),
Preferences(idUser, idBird, activePreference),
Mentions(idPost, idUser, activeMention),
Tags(idPost, idBird, activeTag),
Views(device, viewDate, browser, IP, idUser, idPost)
```

Likes(idUser, idPost, likeType, activeLike)

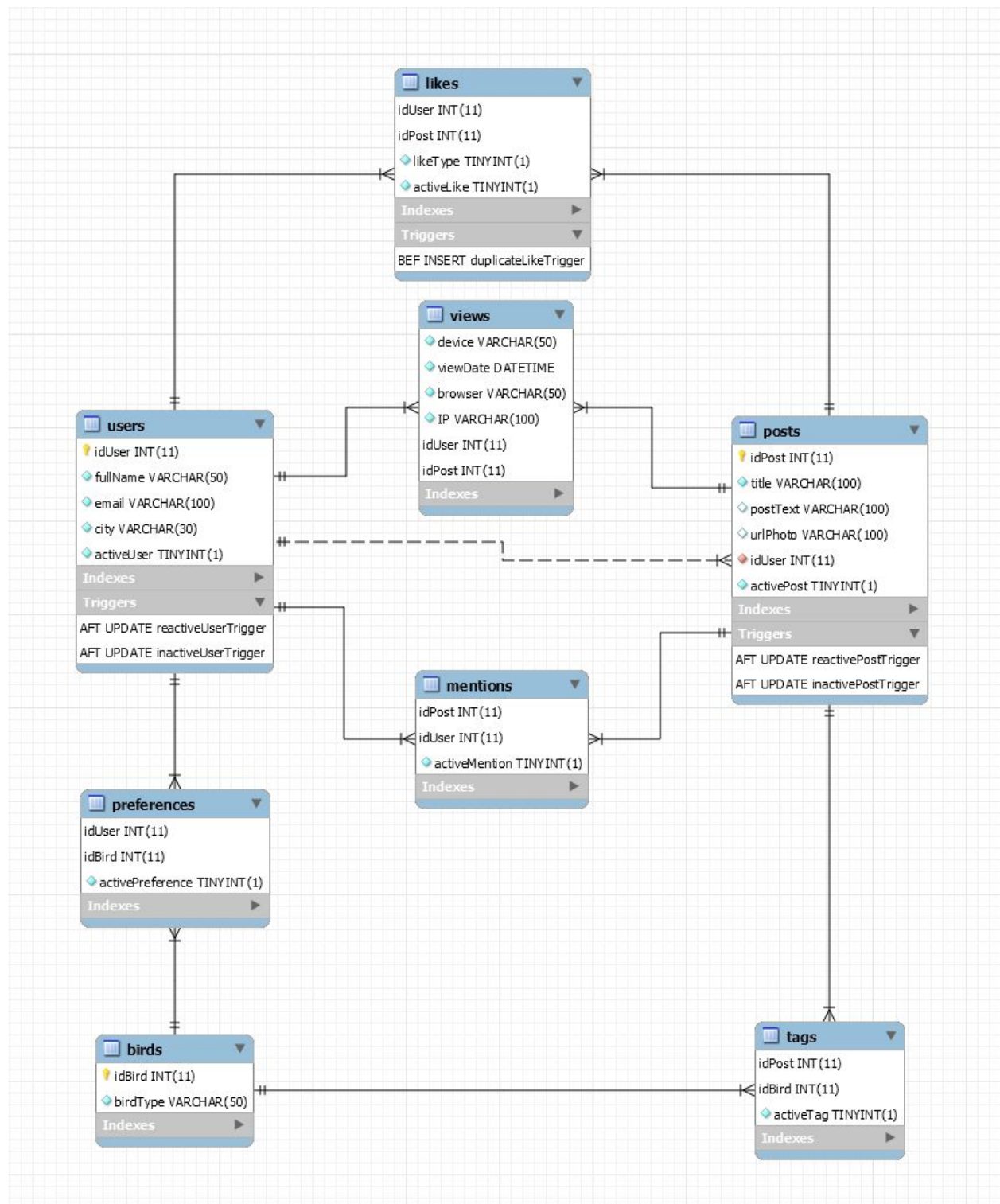
Dicionário de dados

O que representa?	Nome do campo	Descrição	Auto-gerada, Como?	PK	FK Target	Restrições
Users	idUser (INT)	Número identificador do usuário	AutoIncrement	Sim		Not Null
	fullName (VARCHAR)	Nome do usuário	-----			Not Null
	email (VARCHAR)	Email do usuário	-----			Not Null
	city (VARCHAR)	Cidade onde o usuário mora	-----			Not Null
	activeUser (TINYINT)	Se o usuário está ativo	Default = 1			Not Null
Posts	idPost (INT)	Número identificador da postagem	AutoIncrement	Sim		Not Null
	title (VARCHAR)	Título da postagem	-----			Not Null
	postText (VARCHAR)	Texto da postagem	-----			
	urlPhoto (VARCHAR)	URL que referencia a foto da postagem	-----			
	idUser (INT)	Número identificador do usuário	Foreign Key		Users	Not Null
	activePost (TINYINT)	Se a postagem está ativa	Default = 1			Not Null
Birds	idBird (INT)	Número identificador	AutoIncrement	Sim		Not Null

		do pássaro				
	birdType (VARCHAR)	Espécie de pássaro	-----			Not Null, Unique
Preferences	idUser (INT)	Número identificador do usuário	Foreign Key	Sim	Users	Not Null
	idBird (INT)	Número identificador do pássaro	Foreign Key	Sim	Birds	Not Null
	activePreference (TINYINT)	Se a preferência é ativa	Default = 1			Not Null
Mentions	idUser (INT)	Número identificador do usuário	Foreign Key	Sim	User	Not Null
	idPost (INT)	Número identificador da postagem	Foreign Key	Sim	Posts	Not Null
	activeMention (TINYINT)	Se a menção está ativa	Default = 1			Not Null
Tags	idPost (INT)	Número identificador da postagem	Foreign Key	Sim	Posts	Not Null
	idBird (INT)	Número identificador do pássaro	Foreign Key	Sim	Birds	Not Null
	activeTag (TINYINT)	Se a “etiqueta” está ativa	Default = 1			Not Null
Views	device (VARCHAR)	Dispositivo no qual foi visualizado	-----			Not Null
	viewDate (DATETIME)	Data e hora no qual foi visualizado	Date Time			Not Null
	browser	Browser no	-----			Not Null

	(VARCHAR)	qual foi visualizado				
	IP (VARCHAR)	IP público do dispositivo	-----			Not Null
	idPost (INT)	Número identificador da postagem	Foreign Key	Sim	Posts	Not Null
	idUser (INT)	Número identificador do usuário	Foreign Key	Sim	Users	Not Null
Likes	idUser (INT)	Número identificador do usuário	Foreign Key	Sim	Posts	Not Null
	idPost (INT)	Número identificador da postagem	Foreign Key	Sim	Users	Not Null
	likeType (TINYINT)	Booleana que indica Joinha ou Anti.	Default = 1			Not Null
	activeLike (TINYINT)	Se a curtida está ativa	Default = 1			Not Null

Diagrama do modelo relacional



Scripts de criação do banco de dados

```

DROP DATABASE IF EXISTS socialNetwork;
CREATE DATABASE socialNetwork;
  
```

```

USE socialNetwork;
CREATE TABLE Users (
    idUser INT NOT NULL AUTO_INCREMENT,
    fullName VARCHAR(50) NOT NULL UNIQUE,
    email VARCHAR(100) NOT NULL UNIQUE,
    city VARCHAR(30) NOT NULL,
    activeUser TINYINT(1) NOT NULL DEFAULT 1,
    PRIMARY KEY (idUser)
);
CREATE TABLE Posts (
    idPost INT NOT NULL AUTO_INCREMENT,
    title VARCHAR(100) NOT NULL UNIQUE,
    postText VARCHAR(100),
    urlPhoto VARCHAR(100),
    idUser INT NOT NULL,
    activePost TINYINT(1) NOT NULL DEFAULT 1,
    PRIMARY KEY (idPost),
    FOREIGN KEY (idUser)
    REFERENCES Users (idUser) ON DELETE CASCADE ON UPDATE CASCADE
);
CREATE TABLE Birds (
    idBird INT NOT NULL AUTO_INCREMENT,
    birdType VARCHAR(50) NOT NULL UNIQUE,
    PRIMARY KEY (idBird)
);
CREATE TABLE Preferences (
    idUser INT NOT NULL,
    idBird INT NOT NULL,
    activePreference TINYINT(1) NOT NULL DEFAULT 1,
    PRIMARY KEY (idUser, idBird),
    FOREIGN KEY (idUser)
    REFERENCES Users (idUser) ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (idBird)
    REFERENCES Birds (idBird) ON DELETE CASCADE ON UPDATE CASCADE
);
CREATE TABLE Mentions (
    idPost INT NOT NULL,
    idUser INT NOT NULL,
    activeMention TINYINT(1) NOT NULL DEFAULT 1,
    PRIMARY KEY (idPost, idUser),
    FOREIGN KEY (idPost)
    REFERENCES Posts (idPost) ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (idUser)
    REFERENCES Users (idUser) ON DELETE CASCADE ON UPDATE CASCADE
);
CREATE TABLE Tags (
    idPost INT NOT NULL,

```

```

        idBird INT NOT NULL,
        activeTag TINYINT(1) NOT NULL DEFAULT 1,
        PRIMARY KEY (idPost, idBird),
        FOREIGN KEY (idPost)
        REFERENCES Posts (idPost) ON DELETE CASCADE ON UPDATE CASCADE,
        FOREIGN KEY (idBird)
        REFERENCES Birds (idBird) ON DELETE CASCADE ON UPDATE CASCADE
    );
CREATE TABLE Views (
    device VARCHAR(50) NOT NULL,
    viewDate DATETIME NOT NULL,
    browser VARCHAR(50) NOT NULL,
    IP VARCHAR(100) NOT NULL,
    idUser INT NOT NULL,
    idPost INT NOT NULL,
    PRIMARY KEY (idUser, idPost),
    FOREIGN KEY (idUser)
    REFERENCES Users (idUser) ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (idPost)
    REFERENCES Posts (idPost) ON DELETE CASCADE ON UPDATE CASCADE
);

```

Script Delta de alteração do banco de dados e criação de um novo trigger prevenindo likes duplicados.

```

USE socialNetwork;
DROP TABLE IF EXISTS Likes;
CREATE TABLE Likes (
    idUser INT NOT NULL,
    idPost INT NOT NULL,
    likeType TINYINT(1) NOT NULL DEFAULT 1,
    activeLike TINYINT(1) NOT NULL DEFAULT 1,
    PRIMARY KEY (idUser, idPost),
    FOREIGN KEY (idUser)
    REFERENCES Users (idUser) ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (idPost)
    REFERENCES Posts (idPost) ON DELETE CASCADE ON UPDATE CASCADE
);

DROP TRIGGER IF EXISTS duplicateLikeTrigger;
DELIMITER //
CREATE TRIGGER duplicateLikeTrigger
BEFORE INSERT ON Likes
FOR EACH ROW
BEGIN
    SELECT COUNT(*) INTO @rowcount FROM Likes WHERE idUser=NEW.idUser AND
    idPost=NEW.idPost;

```

```

        IF @rowcount > 0 THEN
            SIGNAL SQLSTATE '45000'
            SET MESSAGE_TEXT = 'Duplicate Like';
        END IF;
    END //
DELIMITER ;

```

Script de criação de triggers e outros constraints para garantir a integridade do banco de dados

```

DELIMITER ;

DROP TRIGGER IF EXISTS reactiveUserTrigger;
DELIMITER //
CREATE TRIGGER reactiveUserTrigger
AFTER UPDATE ON Users
FOR EACH ROW
BEGIN
    IF NEW.activeUser = 1 THEN
        UPDATE Posts SET activePost = 1
        WHERE NEW.idUser = Posts.idUser;
        UPDATE Mentions SET activeMention = 1
        WHERE NEW.idUser = Mentions.idUser;
        UPDATE Preferences SET activePreference = 1
        WHERE NEW.idUser = Preferences.idUser;
        UPDATE Likes SET activeLike = 1
        WHERE NEW.idUser = Likes.idUser;
    END IF;
END; //
DELIMITER ;

DROP TRIGGER IF EXISTS reactivePostTrigger;
DELIMITER //
CREATE TRIGGER reactivePostTrigger
AFTER UPDATE ON Posts
FOR EACH ROW
BEGIN
    IF NEW.activePost = 1 THEN
        UPDATE Mentions SET activeMention = 1
        WHERE NEW.idPost = Mentions.idPost;
        UPDATE Tags SET activeTag= 1
        WHERE NEW.idPost = Tags.idPost;
        UPDATE Likes SET activeLike= 1
        WHERE NEW.idPost = Likes.idPost;
    END IF;
END; //
DELIMITER ;

```



```
        END IF;  
END; //  
DELIMITER ;
```