



This is the web_page link: [er model link](#)

-- Here, we use auto-increment for the ID. Later, we will consider using the Snowflake algorithm or other methods to generate IDs.

```
sql

-- Here, we use auto-increment for the ID. Later, we will consider using the Snowflake
algorithm or other methods to generate IDs.

-- Create Database
CREATE DATABASE IF NOT EXISTS social_media_analysis;
USE social_media_analysis;

-- Create Social Media Platform Table
CREATE TABLE social_media (
    media_id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(50) NOT NULL UNIQUE
) COMMENT 'Stores social media platform information';

-- Create User Table
CREATE TABLE user (
    user_id INT AUTO_INCREMENT PRIMARY KEY,
    media_id INT NOT NULL,
    username VARCHAR(40) NOT NULL,
    first_name VARCHAR(50),
    last_name VARCHAR(50),
    country_of_birth VARCHAR(50),
    country_of_residence VARCHAR(50),
    age INT,
    gender VARCHAR(20),
    is_verified BOOLEAN DEFAULT FALSE,
    FOREIGN KEY (media_id) REFERENCES social_media(media_id),
    UNIQUE KEY (media_id, username) -- Username must be unique on a specific platform
) COMMENT 'Stores social media user information';

-- Create Post Table
CREATE TABLE post (
    post_id INT AUTO_INCREMENT PRIMARY KEY,
    user_id INT NOT NULL,
    media_id INT NOT NULL,
    content TEXT NOT NULL,
    posted_time DATETIME NOT NULL,
    location_city VARCHAR(50),
    location_state VARCHAR(50),
    location_country VARCHAR(50),
    likes INT DEFAULT 0,
    dislikes INT DEFAULT 0,
    has_multimedia BOOLEAN DEFAULT FALSE,
    FOREIGN KEY (user_id) REFERENCES user(user_id),
    FOREIGN KEY (media_id) REFERENCES social_media(media_id),
    UNIQUE KEY (user_id, media_id, posted_time) -- A user cannot post multiple times at the
same time on the same platform
) COMMENT 'Stores social media posts';
```

sql

```
-- Create Repost Table
CREATE TABLE repost (
    repost_id INT AUTO_INCREMENT PRIMARY KEY,
    post_id INT NOT NULL,
    user_id INT NOT NULL,
    repost_time DATETIME NOT NULL,
    FOREIGN KEY (post_id) REFERENCES post(post_id),
    FOREIGN KEY (user_id) REFERENCES user(user_id)
) COMMENT 'Stores repost information';

-- Create Research Institute Table
CREATE TABLE institute (
    institute_id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(100) NOT NULL UNIQUE
) COMMENT 'Stores research institute information';

-- Create Project Table
CREATE TABLE project (
    project_id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(100) NOT NULL UNIQUE,
    manager_first_name VARCHAR(50) NOT NULL,
    manager_last_name VARCHAR(50) NOT NULL,
    institute_id INT NOT NULL,
    start_date DATE NOT NULL,
    end_date DATE NOT NULL,
    FOREIGN KEY (institute_id) REFERENCES institute(institute_id),
    CONSTRAINT chk_date CHECK (end_date >= start_date)
) COMMENT 'Stores analysis project information';

-- Create Project Field Table
CREATE TABLE project_field (
    field_id INT AUTO_INCREMENT PRIMARY KEY,
    project_id INT NOT NULL,
    field_name VARCHAR(50) NOT NULL,
    FOREIGN KEY (project_id) REFERENCES project(project_id),
    UNIQUE KEY (project_id, field_name) -- Field name must be unique within a project
) COMMENT 'Stores project analysis fields';
```

sql

```
- Create Project-Post Association Table
CREATE TABLE project_post (
    project_post_id INT AUTO_INCREMENT PRIMARY KEY,
    project_id INT NOT NULL,
    post_id INT NOT NULL,
    FOREIGN KEY (project_id) REFERENCES project(project_id),
    FOREIGN KEY (post_id) REFERENCES post(post_id),
    UNIQUE KEY (project_id, post_id) -- Avoid duplicate analysis
) COMMENT 'Stores the association between projects and posts';

-- Create Analysis Result Table
CREATE TABLE analysis_result (
    result_id INT AUTO_INCREMENT PRIMARY KEY,
    project_post_id INT NOT NULL,
    field_id INT NOT NULL,
    value VARCHAR(255),
    FOREIGN KEY (project_post_id) REFERENCES project_post(project_post_id),
    FOREIGN KEY (field_id) REFERENCES project_field(field_id),
    UNIQUE KEY (project_post_id, field_id) -- Each post has only one result per field
) COMMENT 'Stores analysis results';

-- Create Indexes to Optimize Query Performance
CREATE INDEX idx_post_media ON post(media_id);
CREATE INDEX idx_post_time ON post(posted_time);
CREATE INDEX idx_post_user ON post(user_id);
CREATE INDEX idx_user_names ON user(first_name, last_name);
CREATE INDEX idx_project_post_project ON project_post(project_id);
CREATE INDEX idx_project_post_post ON project_post(post_id);
CREATE INDEX idx_analysis_result_field ON analysis_result(field_id);
```