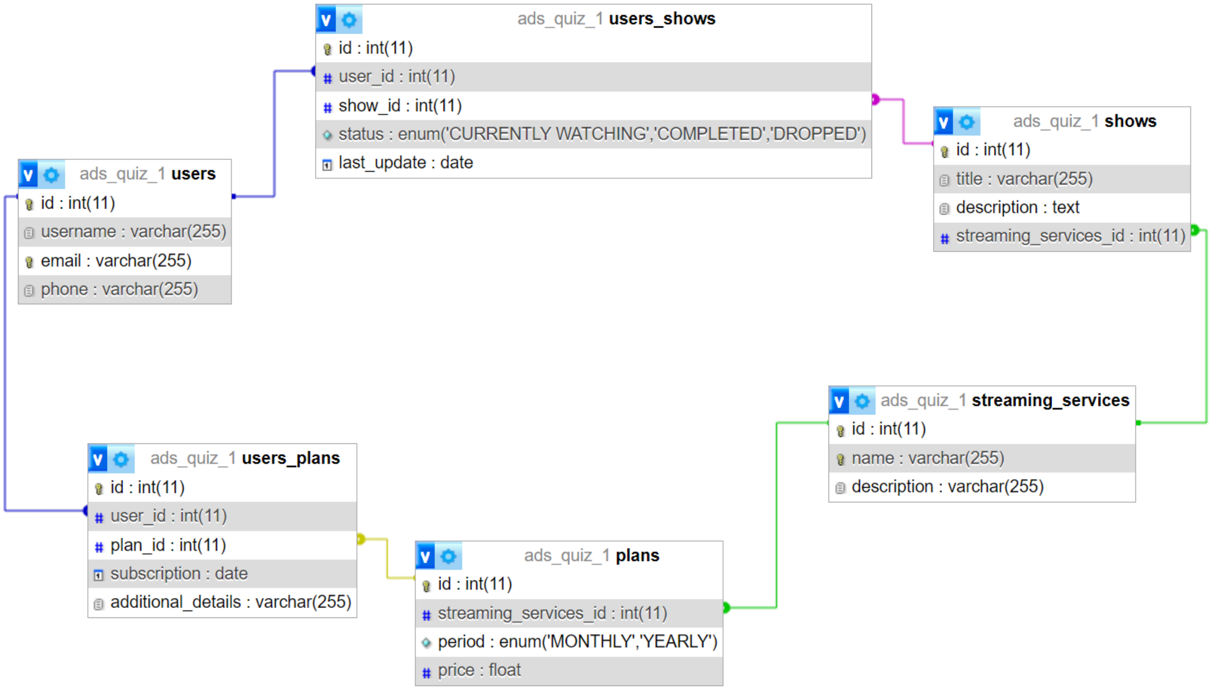
```sql
-- Drop tables that reference others first
DROP TABLE IF EXISTS users_shows;
DROP TABLE IF EXISTS users_plans;

-- Drop tables that are referenced by others
DROP TABLE IF EXISTS shows;
DROP TABLE IF EXISTS plans;

-- Drop the remaining tables
DROP TABLE IF EXISTS streaming_services;
DROP TABLE IF EXISTS users;

CREATE TABLE users (
    id INT PRIMARY KEY AUTO_INCREMENT,
    username VARCHAR(255),
    email VARCHAR(255) NOT NULL UNIQUE,
    phone VARCHAR(255)
);

CREATE TABLE streaming_services (
    id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(255) NOT NULL UNIQUE,
    description VARCHAR(255)
);

CREATE TABLE plans (
    id INT PRIMARY KEY AUTO_INCREMENT,
    streaming_services_id INT,
    period ENUM('MONTHLY', 'YEARLY') NOT NULL,
    price FLOAT,
    FOREIGN KEY (streaming_services_id) REFERENCES streaming_services(id)
);

CREATE TABLE users_plans (
    id INT PRIMARY KEY AUTO_INCREMENT,
    user_id INT,
    plan_id INT,
    subscription DATE,
    additional_details VARCHAR(255),
    FOREIGN KEY (user_id) REFERENCES users(id),
    FOREIGN KEY (plan_id) REFERENCES plans(id)
);

CREATE TABLE shows (
    id INT PRIMARY KEY AUTO_INCREMENT,
    title VARCHAR(255) NOT NULL,
    description TEXT,
    streaming_services_id INT,
    FOREIGN KEY (streaming_services_id) REFERENCES streaming_services(id)
);

CREATE TABLE users_shows (
    id INT PRIMARY KEY AUTO_INCREMENT,
    user_id INT,
    show_id INT,
    status ENUM('CURRENTLY WATCHING', 'COMPLETED', 'DROPPED') NOT NULL,
    last_update DATE,
    FOREIGN KEY (user_id) REFERENCES users(id),
    FOREIGN KEY (show_id) REFERENCES shows(id)
);
```



ads_quiz_1 **users_shows**
id : int(11)
user_id : int(11)
show_id : int(11)
status : enum('CURRENTLY WATCHING','COMPLETED','DROPPED')
last_update : date

ads_quiz_1 **shows**
id : int(11)
title : varchar(255)
description : text
streaming_services_id : int(11)

ads_quiz_1 **users**
id : int(11)
username : varchar(255)
email : varchar(255)
phone : varchar(255)

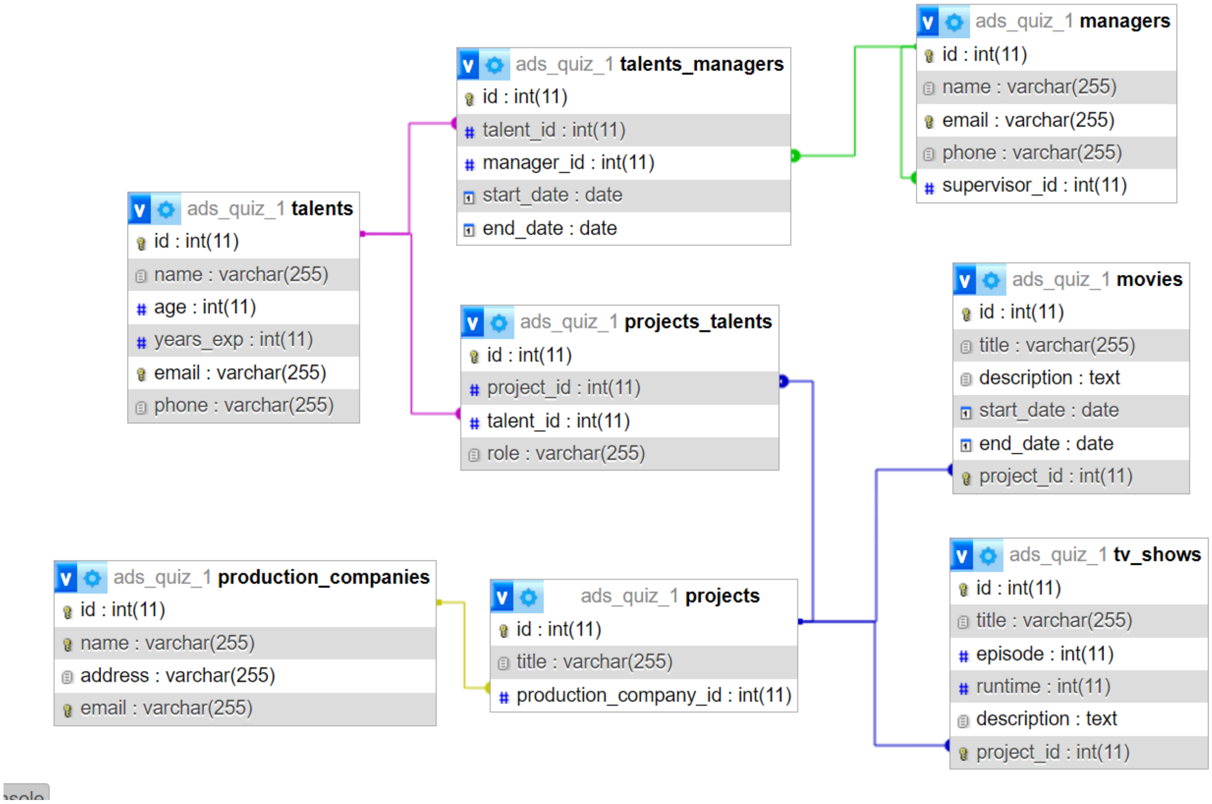
ads_quiz_1 **streaming_services**
id : int(11)
name : varchar(255)
description : varchar(255)

ads_quiz_1 **users_plans**
id : int(11)
user_id : int(11)
plan_id : int(11)
subscription : date
additional_details : varchar(255)

ads_quiz_1 **plans**
id : int(11)
streaming_services_id : int(11)
period : enum('MONTHLY','YEARLY')
price : float

```
-- Creating production_companies table first
CREATE TABLE production_companies (
    id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(255) NOT NULL UNIQUE,
    address VARCHAR(255),
    email VARCHAR(255) NOT NULL UNIQUE
);
```

```
-- Creating managers table next
CREATE TABLE managers (
   id INT PRIMARY KEY AUTO_INCREMENT,
   name VARCHAR(255) NOT NULL,
   email VARCHAR(255) UNIQUE,
   phone VARCHAR(255),
   supervisor_id INT,
   FOREIGN KEY (supervisor_id) REFERENCES managers(id) ON DELETE SET NULL --
Self-referencing FK
);

-- Creating talents table
CREATE TABLE talents (
   id INT PRIMARY KEY AUTO_INCREMENT,
   name VARCHAR(255) NOT NULL,
   age INT NOT NULL,
   years_exp INT NOT NULL,
   email VARCHAR(255) UNIQUE,
   phone VARCHAR(255)
);

-- Creating talents_managers table
CREATE TABLE talents_managers (
   id INT PRIMARY KEY AUTO_INCREMENT,
   talent_id INT,
   manager_id INT,
   start_date DATE,
   end_date DATE,
   FOREIGN KEY (talent_id) REFERENCES talents(id) ON DELETE CASCADE,
   FOREIGN KEY (manager_id) REFERENCES managers(id) ON DELETE CASCADE
);

-- Creating projects table
CREATE TABLE projects (
   id INT PRIMARY KEY AUTO_INCREMENT,
   title VARCHAR(255) NOT NULL,
   production_company_id INT,
   FOREIGN KEY (production_company_id) REFERENCES production_companies(id) ON
DELETE SET NULL
);

-- Creating projects_talents table
CREATE TABLE projects_talents (
   id INT PRIMARY KEY AUTO_INCREMENT,
   project_id INT,
   talent_id INT,
   role VARCHAR(255), -- Role of the talent in the project
   FOREIGN KEY (project_id) REFERENCES projects(id) ON DELETE CASCADE,
   FOREIGN KEY (talent_id) REFERENCES talents(id) ON DELETE CASCADE
);

-- Creating tv_shows table
CREATE TABLE tv_shows (
   id INT PRIMARY KEY AUTO_INCREMENT,
   title VARCHAR(255) NOT NULL,
   episode INT NOT NULL,
   runtime INT NOT NULL,
   description TEXT NOT NULL,
   project_id INT UNIQUE,
   FOREIGN KEY (project_id) REFERENCES projects(id) ON DELETE CASCADE
);

-- Creating movies table
CREATE TABLE movies (
   id INT PRIMARY KEY AUTO_INCREMENT,
   title VARCHAR(255) NOT NULL,
   description TEXT NOT NULL,
   start_date DATE,
   end_date DATE,
   project_id INT UNIQUE,
   FOREIGN KEY (project_id) REFERENCES projects(id) ON DELETE CASCADE
);
```



ads_quiz_1 **talents**

- id : int(11)
- name : varchar(255)
- age : int(11)
- years_exp : int(11)
- email : varchar(255)
- phone : varchar(255)

ads_quiz_1 **talents_managers**

- id : int(11)
- talent_id : int(11)
- manager_id : int(11)
- start_date : date
- end_date : date

ads_quiz_1 **managers**

- id : int(11)
- name : varchar(255)
- email : varchar(255)
- phone : varchar(255)
- supervisor_id : int(11)

ads_quiz_1 **projects_talents**

- id : int(11)
- project_id : int(11)
- talent_id : int(11)
- role : varchar(255)

ads_quiz_1 **movies**

- id : int(11)
- title : varchar(255)
- description : text
- start_date : date
- end_date : date
- project_id : int(11)

ads_quiz_1 **production_companies**

- id : int(11)
- name : varchar(255)
- address : varchar(255)
- email : varchar(255)

ads_quiz_1 **projects**

- id : int(11)
- title : varchar(255)
- production_company_id : int(11)

ads_quiz_1 **tv_shows**

- id : int(11)
- title : varchar(255)
- episode : int(11)
- runtime : int(11)
- description : text
- project_id : int(11)