Assigment2

Written Schemas and Scenario description:

Foreign keys for table schema can be found below

ER diagrams and table schemas can also be found below in order

Problem 1:

```
Create table Movies(
Movie_id int not null primary key, ---movie identifier
sp_id int not null references Scenes(s_id),  -- the screenplay the move was based on
foreign key(sp_id)
);
Create table screenplay(
sp_id int not null primary key, ---screenplay identifier
title varchar(30 ) NOT NULL, --title of the screen play
author varchar(30 ) NOT NULL, --author if the screen play
);
Create table scenes(
s_id int not null primary key, ---scene identifier
slocation varchar(30 ) NOT NULL, --location where the scene is set to take place.
flocation varchar(30 ) NOT NULL, --location where the scene is filmed irl.
sp  int not null references screenplay(s_id),
Foreign key(sp_id)
);
Create table actor(
name varchar(30 )  not null primary key, ---name of actor
a_name varchar(30 )  , ---name of actor's agent
```

```
address varchar(30 ) NOT NULL, --actor's home location
pnumber varchar(30 ) NOT NULL, --actor's phone number
);

Create table acted(
name varchar(30)  not null references actor (name), ---name of actor in s_id
s_id int not null references scene (s_id), ---scene with actor name
primary key(name, s_id),
foreign key(name, s_id)
);
Create table taken(
takenum int not null, ---number of takes.
s_id int not null primary key references scene (s_id), ---scene with actor name
primary key(takenum, s_id)
);
```

Problem 2:

```
create table recordings(
id int primary key not null,
 title varchar(30 ) NOT NULL,
artist varchar(30) NOT NULL,
 producer varchar(30)  default NULL,
```

```
 year int );

create table tracks (
albumID int NOT NULL references recordings(id),
 songID int NOT NULL references songs(sid),
 track_number int NOT NULL,
foreign key (albumID,songID),

primary key (albumID,songID) );

create table users (
user_id varchar(30) primary key not null,
name varchar(30) not null );

create table playlists(
user_id varchar(30)
not null references users(user_id),
 playlist_name varchar(30) not null,
foreign key (user_id),
primary key (user_id,playlist_name) );

create table playlist_tracks(
user_id varchar(30) not null,
 playlist_name varchar(30) not null,
 song_id int not null references songs (sid),
```

```
 primary key (user_id, playlist_name,song_id),

foreign key (user_id, playlist_name, song_id)

references playlists(user_id, playlist_name) );


create table songs (

sid int primary key,

title varchar(30) NOT NULL,

authors varchar(30) NOT NULL,

);

create table audio (

sid int primary key reference song(sid),

type varchar(30) NOT NULL –the type of audio recording mp3 etc

);
```

since audio is inheriting from songs, I have decided to use the profs second style of representing inheritance.

Problem 3:

```
create table book(

book_id int primary key not null, -- unique identifier for books

 title varchar(30 ) NOT NULL, -- this is the title of the book

publisher varchar(30) NOT NULL, --this is the publisher of the book

date  varchar(30)  not NULL, --this is the date of publication of the book

 year int

offset varchar(30), -- this is the difffrence between the users page number and the page the valsue actually appears in the book

 );
```

```
create table user(
user_id int primary key not null, -- unique identifier for a user
 password varchar(30 ) NOT NULL -- the users login password
);
create table upload(
user int not null reference to user(user_id), -- unique identifier for a user the uploads book
book  varchar(30 ) NOT NULL reference to book(book_id), -- the users login password
primary key (user, book),
foreign key (user, book)
);
```

For this ER model I decided to make the relationship between book and user into a table because I made if into an end to end relationship.

```
create table songs(
song_id int not null primary key, -- unique identifier for a song
title varchar(30 ) NOT NULL, -- this is the title of the song
composer varchar(30 ) NOT NULL, -- this is the composer of the song
pnum varchar(30 ) NOT NULL, -- this is the page the song appears in the book
length varchar(30 ) NOT NULL, -- this is the length of the song in the book
bookid  varchar(30 ) NOT NULL reference to book(book_id), -- the users login password
foreign key ( book)
);
```

Problem 4:

Scenario:

A lot of upcoming designers and fashion students depend on already made patterns, designs, illustration and how to videos to find inspiration to come up with pieces. Students and upcoming designers also create their own patterns, illustrations, and instructional videos which they can upload into the application. Users will be able to subscribe the other users and have user's subscribers to them, users are also able to rate designs. The database will catalog designs that are uploaded, the designs would have title, upload date, type of upload (patterns, designs, illustration and how to videos), and an average rating of stars (1-5).

A pattern would have the number of sections it takes to form a full piece, and an illustration would have the type of media and any specifications set by the author (the user), the howtovideos contain a difficulty level that rates the content of the video from baby, princess and queen, with baby being the easiest and queen the hardest. The howtovideos should also be able to tel if that video is related to a pattern, illustration or video that is already in the system.

Schema:

```
create table user(

user_id int primary key not null, -- unique identifier for a user

 password varchar(30 ) NOT NULL -- the users login password

subscribers varchar(30) –the user's subscribers

subscriptions varchar(30) –the user's subscriptions


);

create table design(

design_id int primary key not null, -- unique identifier for a user

 title varchar(30 ) NOT NULL, -- the users login password

rating varchar(30), –the user's subscribers

udate varchar(30) not null, –the user's subscriptions

style varchar(30) not null,–the style of the design

era varchar(30) not null, – the ear the style is from

author varchar(30) not null – this will usually be the user who uploaded the material

);
```

```
Create table upload(

user_id int not null reference user(user_id), -- user id of the user that uploaded design_id

design_id int not null reference design(design_id), -- the design that user_id uploaded

primary key(user_id,design_id),

foreign key(user_id,design_id)


);


create table pattern(

design_id int not null reference design(design_id) primary key, -- design id to show that pattern is a design

nosection varchar(30) not null, –the number of sections that the pattern has

tool varchar(30) not null –any special tools needed to create pattern


);

create table illustration(

design_id int not null reference design(design_id) primary key, -- design id to show that illustration is a design

 type varchar(30 ) NOT NULL, – the type of media ie pdf, .jpg, .png etc

extra  varchar(30 ) – this any extra information about the illustration such as clour, material etc


);

create table how-to-video(

design_id int not null reference design(design_id) primary key, -- design id to show that how-to-video is a design

 length  varchar(30 ) NOT NULL, – length of the video

difficulty  varchar(30 ) NOT NULL, – the difficulty  of the material in the video
```

rdesign_id int reference design(design_id)-- if the information referse to any other existing design

);

problem 1

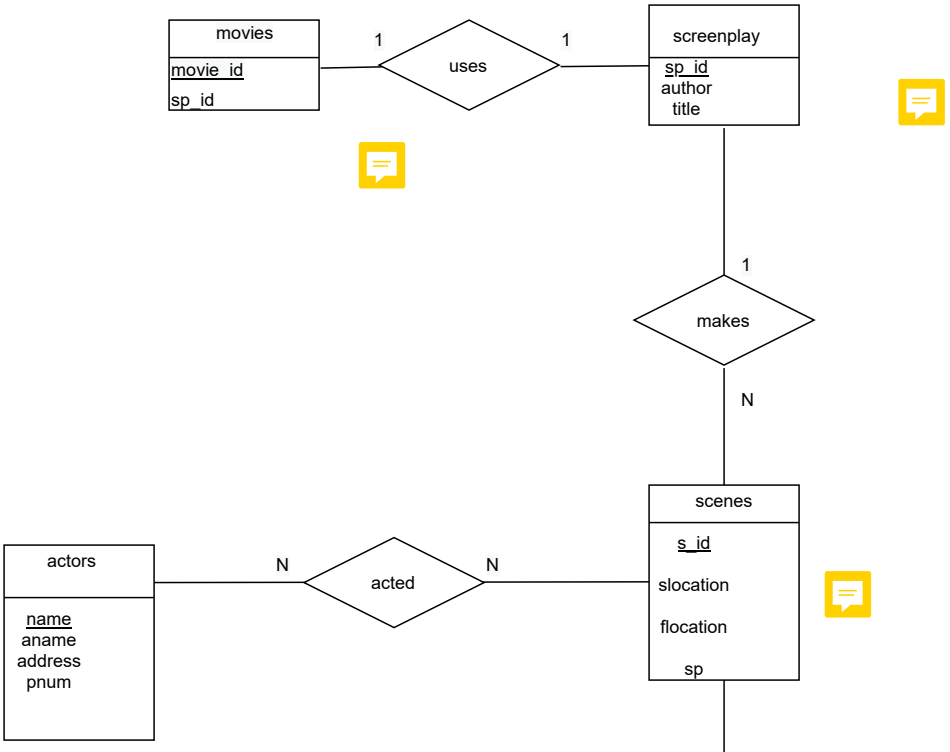


table Schema

movies

movie_id	sp_id

screenplay

sp_id	author	title

scenes

s_id	sp	slocation	flocatiion

take

s_id	takenumber

actors

name	sname	address	pnum

acted

s_id	name

problem 2

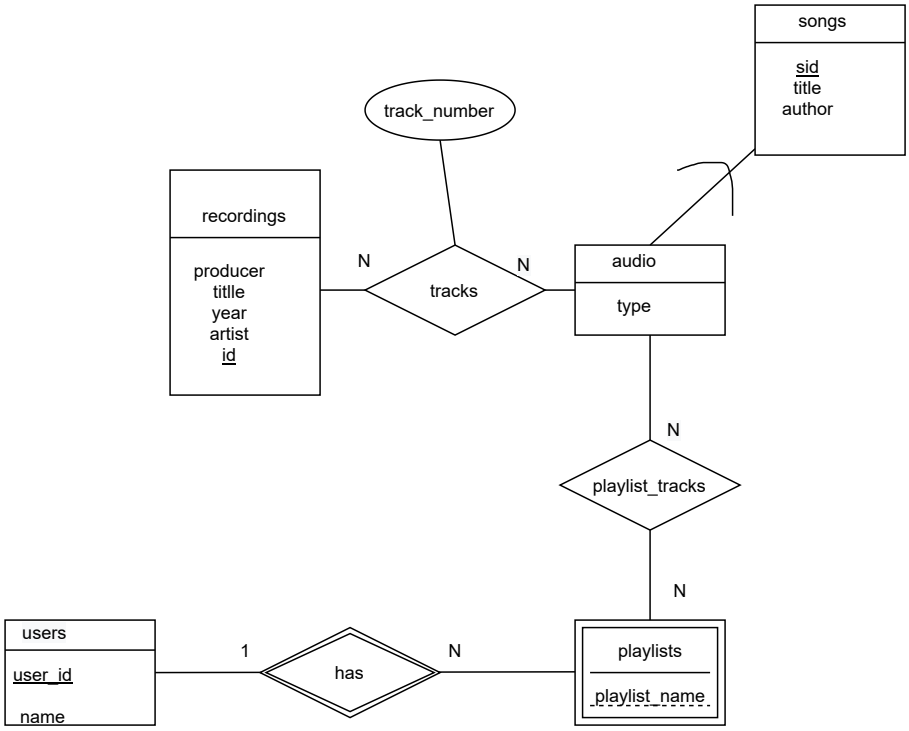


Table Schema

recordings

id	title	producer	artist	year

user

user_id	name

tracks

albumID	songID	track_number

playlists

user_id	playlist_name

songs

sid	title	author

playlist_tracks

user_id	name	playlist_name

audio

sid	type

Table Schema

Book

book_id	title	publisher	date	offset

users

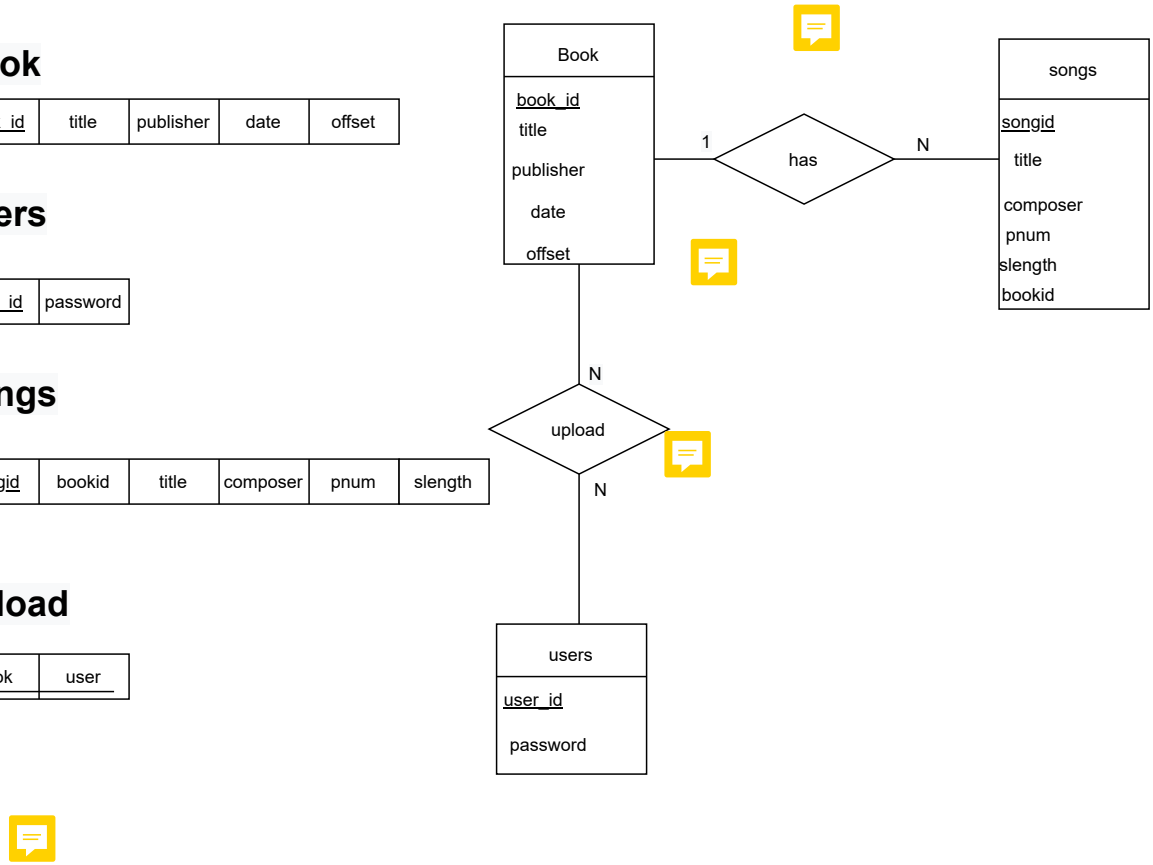
user_id	password

songs

songid	bookid	title	composer	pnum	slength

upload

book	user



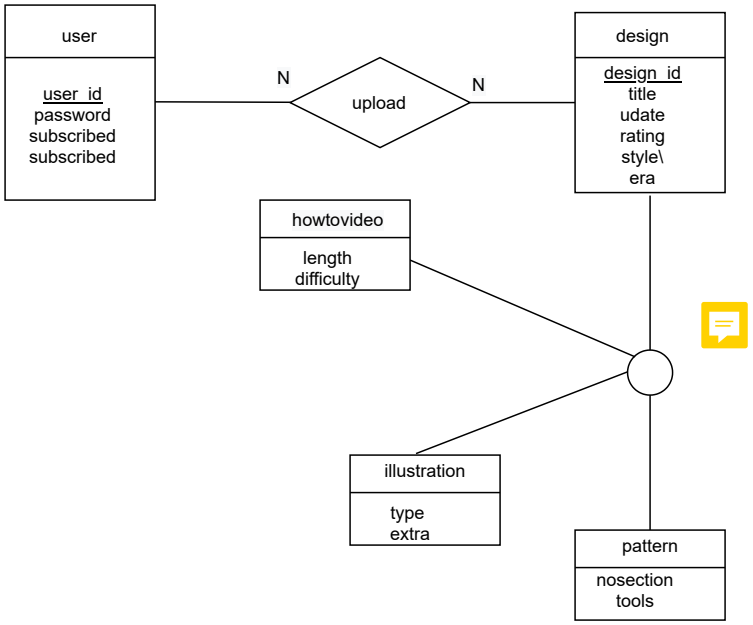


Table Schema

user

user_id	password	subscribed	subscribers

design

design_id	title	udate	rating	style	era

pattern

desig_id	nosection	tools

illustration

desig_id	type	extra

howtovideo

desig_id	length	difficulty

upload

user_id	design_id