

CS-GY 6083 A: Principles of Database Systems

Project: Quiz Management System

(Port #: 8676)

Link: <http://jedi.poly.edu:8676/>

Jay Sarvaiya (js12178)

Priyank Viradia (pdv8883)

Instructor: Julia Stoyanovich

Project Description

- We have developed an online movie ticket booking system. We have implemented the system with keeping in mind 2 types of user interactions which are user interaction and employee interaction. Here a user is a normal user who can check on the Web UI movies that are being currently screened, what are the movie timings etc. and accordingly decide if they wish to watch that movie or not. Employee interactions involve deciding which movie to screen next and basically manage the movie being screened in the cinema screen.
- Lists of entity sets
 - Employee
 - User
 - MoviePool
 - Screen
 - Seats
 - ScreeningMovie
 - Ticket
- Relationship Between Entities
 - Manages
 - Reserved
 - Contains
 - Used_in
 - has
 - played at

Business Rules

→Users are uniquely identified by their user ids

→ Tickets are uniquely identified by the ticket id. A user can reserve any number of tickets, but each ticket is reserved by exactly one user. Each ticket contains exactly one seat_id and the movie they are watching (Screeningmovie).

→ Moviepool contains list of all the movies in the database. They are uniquely identified by the Movie_id and M_name (Movienamename) .All the movies in the moviepool entity are managed by Employee (Total participation between Moviepool entity and manages relation).

→ Screen is uniquely identified by Sc_id (screen id).

→ Seats is weak entity set and it depends on the strong entity Screen. It means that if there are no screens in the cinema there will be no seats. Seats are uniquely identified by seat_id and sc_id.

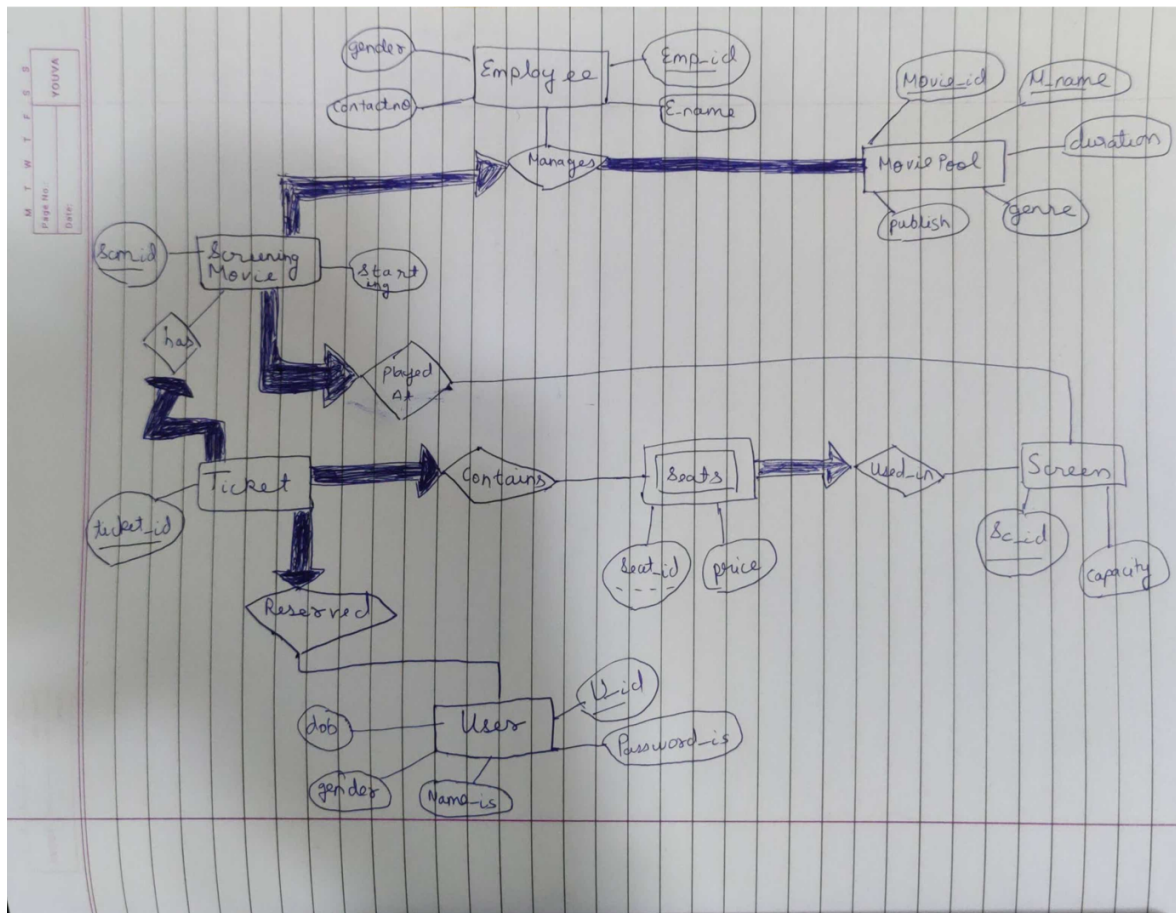
→ ScreeningMovie entity contains the list of movie that are being currently screened in the cinema. Each screening movie is selected by exactly one employee from the moviepool. At a given time exactly one movie is played at any given screen. It is uniquely identified by scm_id (screening movie id).

→ Employee are uniquely identified by their emp_id.

How did we acquire the data?

→ For Screen entity we have written insert into statements and for the rest of the entities we retrieved the data from the website <https://www.mockaroo.com/>. In the form of csv files with respect to each entity and manually checked the .csv files to make sure the data is consistent and correct. We imported the .csv files to populate our database.

ER DIAGRAM



Schema SQL:

```
drop table if exists Employee cascade;
drop table if exists MoviePool cascade;
drop table if exists ScreeningMovie cascade;

drop table if exists Seats cascade;
drop table if exists Users cascade;
drop table if exists Ticket cascade;

create table Screen (
  Sc_id varchar(32) primary key,
  capacity integer
);

insert into Screen(Sc_id,capacity) values('A1',10);
insert into Screen(Sc_id,capacity) values('A2',15);
insert into Screen(Sc_id,capacity) values('A3',10);
insert into Screen(Sc_id,capacity) values('A4',10);
insert into Screen(Sc_id,capacity) values('A5',9);
insert into Screen(Sc_id,capacity) values('B1',10);
insert into Screen(Sc_id,capacity) values('B2',7);
insert into Screen(Sc_id,capacity) values('B3',8);
insert into Screen(Sc_id,capacity) values('B4',20);
insert into Screen(Sc_id,capacity) values('B5',5);
insert into Screen(Sc_id,capacity) values('C1',9);
insert into Screen(Sc_id,capacity) values('C2',5);
insert into Screen(Sc_id,capacity) values('C3',5);
insert into Screen(Sc_id,capacity) values('C4',5);
insert into Screen(Sc_id,capacity) values('C5',20);
insert into Screen(Sc_id,capacity) values('D1',15);
insert into Screen(Sc_id,capacity) values('D2',10);
insert into Screen(Sc_id,capacity) values('D3',20);
insert into Screen(Sc_id,capacity) values('D4',10);
insert into Screen(Sc_id,capacity) values('D5',10);

create table MoviePool (
  Movie_id integer,
  M_name varchar(128),
  Duration integer not null,
  publish date,
  Genre varchar(128),
  primary key (Movie_id, M_name)
);
```

```

create table Employee (
  Emp_id integer primary key,
  E_name varchar(32),
  Gender varchar(32),
  contact_no bigint
);

create table Users (
  U_id varchar(32) primary key,
  Name_is varchar(32),
  password_is varchar(128),
  Gender varchar(32),
  Dob date
);

create table ScreeningMovie (
  Scm_id varchar(32) primary key,
  Movie_id integer not null,
  M_name varchar(128) not null ,
  starting time,
  Emp_id integer not null,
  Sc_id varchar(32) not null,
  foreign key (Movie_id, M_name) references MoviePool (Movie_id, M_name),
  foreign key (Emp_id) references Employee (Emp_id),
  foreign key (Sc_id) references Screen (Sc_id)
);

create table Seats (
  Seat_id varchar(32),
  Sc_id varchar(32),
  price decimal,
  primary key (Seat_id, Sc_id),
  foreign key (Sc_id) references Screen (Sc_id) on delete cascade
);

create table Ticket (
  Ticket_id varchar(32) primary key,
  Seat_id varchar(32) not null,
  Sc_id varchar(32) not null,
  Scm_id varchar(32) not null,
  U_id varchar(32) not null,
  foreign key (Seat_id, Sc_id) references Seats (Seat_id, Sc_id),
  foreign key (U_id) references Users (U_id),
  foreign key (Scm_id) references ScreeningMovie (Scm_id)
);

```

```
);
```