Principles of Databases

Assignment

20191531-Wang Xiang Dong-Jack

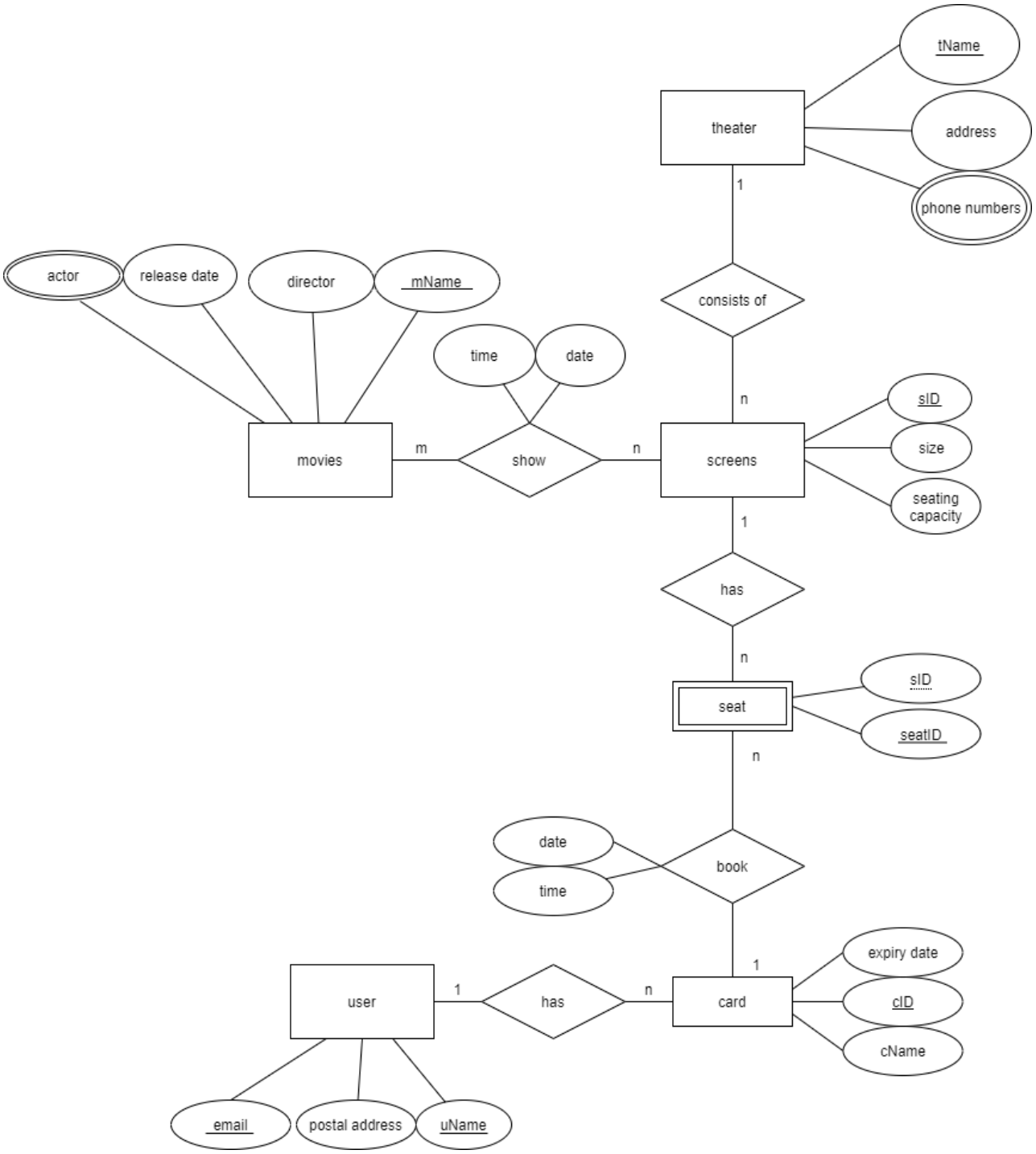
20195534-Lin Jia Zheng-Lam

Content

1.Entity class

- theater: It is uniquely decided by its name and address, and has series of phone number.
- screen: This means the screening room, and it contains the information of size and capacity. And it also show movie at specific date and time.
- seat: This is a weak entity, and it depends on sID of screen, which use to mark seats booked by user.
- movie: The movie has basic information of its director, release time and actors.
- user: Each user can be identified by its name and email, and it also record the postal address of its own.
- card: Every user can have multiple card, so the card is identified by its card-ID, and also has its expiry date and name.

2.Entity-Relationship Diagram



3.Relational Data Model

1.User(uName, email, postal address)

2.card(uName,email,cID,epiry date,cName)

3.The relationship of card-book-seat is translated to UserOrder(cID,sID,seatID,date,time)

4. movie(mName,release time,director,actor)

5. theater(tName,address,phone number)

6. screen(sID,size,capacity)

7. seat(sID,seatID)

8.The relationship of screen-show-movie is translated to session(sID,mName,date,time)

4. Normalization

We decided to use BCNF. After decompositing the relation data model, our normalization as followed:

1.User(uName, email, postal address)

FD={uName,email→postal address}

2.card(cID,epiry date,cName)

FD={cID→epiry date,cName}

3.UC(uName,email,cID)

FD={}

4.movie(mName,release time,director)

FD={mName→release time,director}

5.actor(mName,aName)

FD={}

6.theater(tName,address)

FD={tName→address }

7.TP(tName,phone number)

FD={}

8.screen(sID,size,capacity)

FD={sID→size,capacity}

9.seat(sID,seatID)

FD={}

10.session(sID,mName,date,time)

FD={}

11.UserOrder(cID,sID,seatID,date,time)

FD={}

5.SQL code

5.1 Build Database

```
create database booking_system;
use booking_system;

create table user (
uName varchar(45) not null,
email varchar(45),
postal_address int,
primary key (uName,email)
);

create table card(
cID int not null,
```

```
epiry_date date,
cName varchar(45),
primary key(cID)
);

create table UC(
uName varchar(45) not null,
email varchar(45),
cID int not null,
foreign key(uName,email) references user(uName,email),
foreign key(cID) references card(cID),
primary key(uName,email,cID)
);

create table movie(
mName varchar(45),
release_time date,
director varchar(45),
primary key(mName)
);
```

```
create table actor(
mName varchar(45),
aName varchar(45),
foreign key (mName) references movie(mName),
primary key (mName,aName)
);

create table theater(
tName varchar(45) not null,
address varchar(45),
primary key(tName)
);

create table TP(
tName varchar(45) not null,
phone_number int not null,
foreign key(tName) references theater(tName),
primary key(tName,phone_number)
);

create table screen(
tName varchar(45) not null,
```

```
sID int not null,
size varchar(45),
capacity int,
primary key(sID),
foreign key (tName)references theater(tName)
);

create table seat(
sID int not null,
seatID int not null auto_increment,
foreign key(sID)references screen(sID),
primary key(seatID)
);

create table session(
sID int not null,
mName varchar(45),
SessionDate date,
SessionTime time,
foreign key(sID)references screen(sID),
foreign key(mName)references movie(mName),
primary key(sID,mName,SessionDate,SessionTime)
```

```
);

create table UserOrder(
cID int not null,
sID int not null,
seatID int not null,
UserDate date,
UserTime time,
foreign key(cID)references card(cID),
foreign key(sID,seatID)references seat(sID,seatID),
primary key(cID,sID,seatID,UserDate,Usertime)
);
```

5.2 Populate Data

We did not fill in a large amount of data, but represented a small part of the data.

```
insert into user values('user1','12345@qq.com',110000);
insert into user values('user2','66666@qq.com',110000);

insert into card values(123456,'2022-09-01','China Bank');
insert into card values(666666,'2050-1-1','China Bank');

insert into UC values('user1','12345@qq.com',123456);
insert into UC values('user2','66666@qq.com',666666);

insert into theater values('theater1','street1');
insert into theater values('theater2','street2');
insert into theater values('theater3','street1');

insert into TP values('theater1',12345678);
```

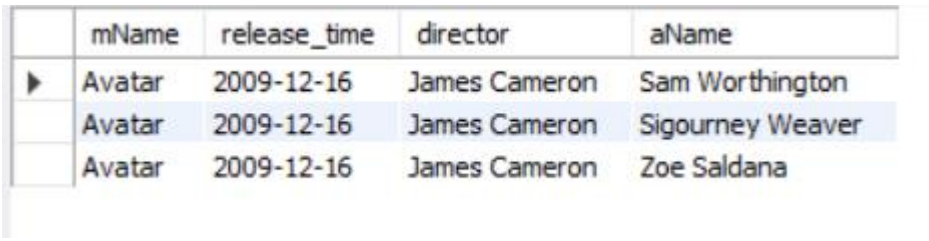
```
insert into TP values('theater1',88888888);
insert into TP values('theater2',66666666);

insert into screen values('theater1',1,'big',120);
insert into screen values('theater1',2,'mid',80);
insert into screen values('theater1',3,'small',60);
insert into screen values('theater2',4,'big',120);
insert into screen values('theater2',5,'mid',80);
insert into screen values('theater2',6,'small',60);

insert into movie values('Avatar','2009-12-16','James Cameron');
insert into movie values('Avengers: Endgame','2019-04-24','Kevin Feige');

insert into actor values('Avatar','Sam Worthington');
insert into actor values('Avatar','Zoe Saldana');
insert into actor values('Avatar','Sigourney Weaver');
insert into actor values('Avengers: Endgame','Robert Downey Jr.');
insert into actor values('Avengers: Endgame','Chris Evans');
```

```
insert into actor values('Avengers: Endgame','Mark Alan Ruffalo');

insert into session values(1,'Avatar','2021-5-2','10:00:00');
insert into session values(1,'Avengers: Endgame','2021-5-2','12:00:00');

insert into seat values(1,1);
insert into seat values(1,2);
insert into seat values(1,3);

insert into userorder values(123456,1,1,'2021-5-2','10:00:00');
insert into userorder values(123456,1,2,'2021-5-2','10:00:00');
insert into userorder values(666666,1,3,'2021-5-2','10:00:00');
```

5.3 Query the Database

```
-- queries base on the file's title: <mName>
select *
from movie m
join actor a
using(mName)
where m.mName='Avatar';
```

result:



mName	release_time	director	aName
Avatar	2009-12-16	James Cameron	Sam Worthington
Avatar	2009-12-16	James Cameron	Sigourney Weaver
Avatar	2009-12-16	James Cameron	Zoe Saldana

```
-- queries base on the director:<director>
select *
from movie m
join actor a
using(mName)
where m.director='James Cameron';
```

result:

mName	release_time	director	aName
Avatar	2009-12-16	James Cameron	Sam Worthington
Avatar	2009-12-16	James Cameron	Sigourney Weaver
Avatar	2009-12-16	James Cameron	Zoe Saldana

-- queries base on the actors:<aName>

```
select *
from movie m
join actor a
using(mName)
where a.aName='Sam Worthington'
```

result:

mName	release_time	director	aName
Avatar	2009-12-16	James Cameron	Sam Worthington