

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC VĂN HIẾN
KHOA CÔNG NGHỆ THÔNG TIN



QUẢN LÝ VÉ XEM PHIM

TIỂU LUẬN MÔN HỌC CƠ SỞ DỮ LIỆU

SVTH: Bùi Minh Nhật

MSSV: 201A290002

GV: Đinh Thị Tâm

Học kỳ I năm học 2021-2022

TP. HỒ CHÍ MINH, 2021

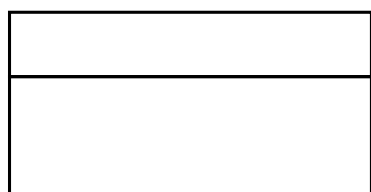
Mục lục

Lưu ý về ký hiệu	ii
Chương I Đặc tả bài toán quản lý vé xem phim	1
1.1 Mẫu vé xem phim	1
1.2 Yêu cầu về mặt kinh doanh	1
1.3 Một số yêu cầu truy vấn	2
Chương II Thiết kế cơ sở dữ liệu	3
2.1 Lược đồ quan hệ	3
2.2 Ràng buộc	4
2.3 Thông tin các bảng	5
2.4 Truy vấn bằng đại số quan hệ	8
Chương III Cài đặt cơ sở dữ liệu	9
3.1 Định nghĩa dữ liệu	9
3.2 Thêm dữ liệu	12
3.3 Truy vấn bằng SQL	16
Chương IV Kết luận	21

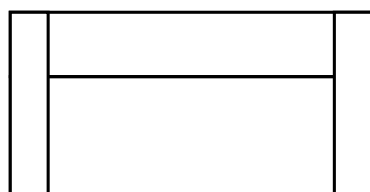
Lưu ý về ký hiệu

Các đoạn mã SQL trình bày trong tiểu luận này phần lớn dựa theo SQL tiêu chuẩn (ISO/IEC 9075-2:2016 / ANSI). Phần mã cài đặt được tác giả tùy chỉnh để chạy trên hệ quản trị cơ sở dữ liệu PostgreSQL 13. Lược đồ thiết kế được ký hiệu bằng Crow's foot notation.

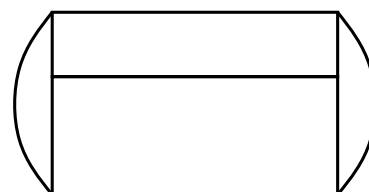
Bảng



Data table



Linking table



Lookup table

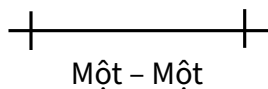
Khóa

PK: Primary key

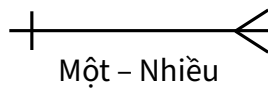
FK: Foreign key

CPK: Composite primary key

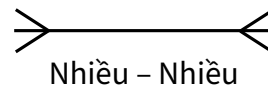
Kiểu quan hệ



Một – Một

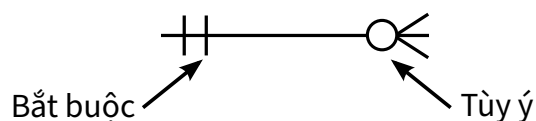


Một – Nhiều



Nhiều – Nhiều

Kiểu tham gia



Quy tắc xóa

(R): Restrict

(C): Cascade

(N): Nullify

Đặc tả bài toán quản lý vé xem phim

Công ty giải trí Con Cò Bé Bé có nhu cầu tin học hóa việc quản lý vé xem phim, tính doanh thu, và thống kê cơ bản. Công ty có các yêu cầu:

1.1 Mẫu vé xem phim

Trên một vé xem phim phải có những thông tin sau đây:

- Mã hóa đơn.
- Ngày giờ đặt vé.
- Chi nhánh rạp.
- Tên phim.
- Ngày chiếu phim.
- Suất chiếu.
- Giá hóa đơn (đã gồm chỗ ngồi).
- Số ghế.
- Nhân viên bán vé.

1.2 Yêu cầu về mặt kinh doanh

1.2.1 Tổ chức rạp

- Công ty có nhiều rạp ở các tỉnh thành. Mỗi tỉnh thành có một rạp.
- Mỗi rạp có một hoặc nhiều nhân viên bán vé.
- Một bộ phim được chiếu nhiều lần: có nhiều suất chiếu trong ngày và chiếu trong nhiều ngày. Ngày giờ chiếu một bộ phim bất kỳ là như nhau ở mọi rạp.
- Mỗi rạp có năm khu chỗ ngồi: A, B, C, D, E. Mỗi khu có các ghế đánh số từ 1, 2, 3,... Các rạp lớn có thể có số ghế ở mỗi khu nhiều hơn

1.2.2 Thông tin phim

- Thông tin cần lưu trữ về phim bao gồm tên phim, đạo diễn, thể loại, năm phát hành, thời lượng, và giá vé niêm yết của phim đó. Những bộ phim có nhiều

phần (series) thì xem mỗi phần là một bộ phim riêng biệt. (Đây là một số thông tin cần cho việc in hóa đơn và thống kê cơ bản. Việc quản lý phim chi tiết nằm ngoài phạm vi của tiểu luận.)

- Mỗi phim chỉ thuộc về một thể loại. Trong một thể loại có thể có nhiều phim.

1.2.3 Khách hàng

- Khách hàng muốn đặt vé trực tuyến phải có tài khoản, quản lý bằng số điện thoại và mật khẩu.
- Khách hàng có thể đặt nhiều chỗ cho một suất chiếu tại cùng một rạp. Mỗi chỗ được tính là một vé.

1.2.4 Giá vé

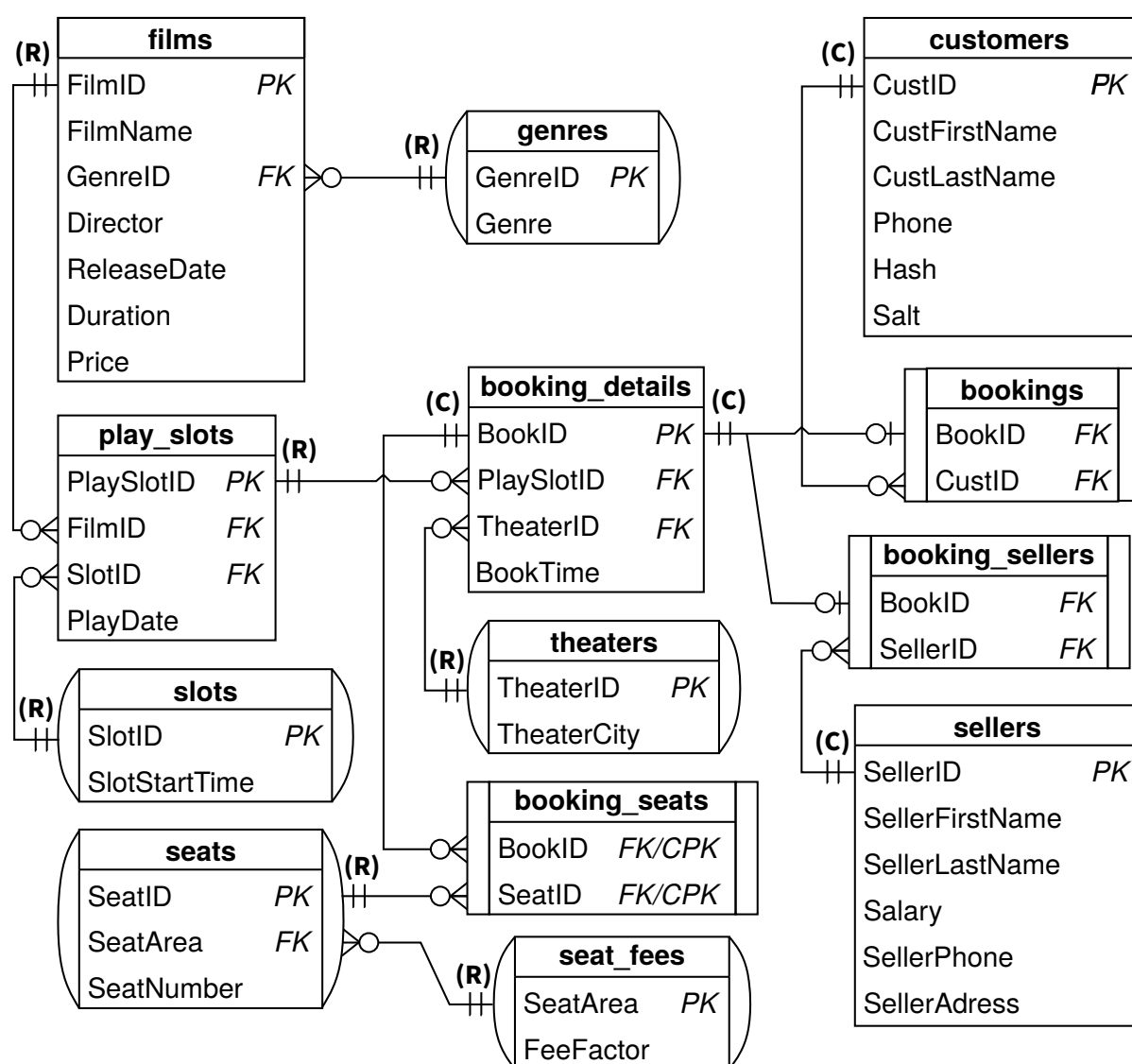
- Giá vé niêm yết do nhà phát hành phim đề xuất. Đơn vị tính là đồng.
- Chỗ ngồi khu A có giá cao hơn 20% giá đề xuất, tương tự khu B là 15%, khu C là 10%, khu D là 5%, khu E bằng với giá đề xuất.

1.3 Một số yêu cầu truy vấn

1. Cho biết các bộ phim thuộc thể loại *Comedy*.
2. Cho biết các bộ phim được chiếu trong tháng 11.
3. Cho biết tỉ lệ đặt vé trực tuyến.
4. Liệt kê những khách hàng đi xem nhiều hơn một lần trong tháng 11.
5. Cho biết các phim chiếu vào khung giờ 20h - 23h (phân biệt với suất chiếu).
6. Tính giá hóa đơn có mã là *tl1m4RRHSj47luJp*.
7. Cho biết số lượng vé được bán tại từng rạp
8. Cho biết tổng doanh thu toàn bộ rạp.
9. Cho biết phim mang lại nhiều doanh thu nhất.
10. Cho biết nhân viên nào bán được nhiều vé nhất.

Thiết kế cơ sở dữ liệu

2.1 Lược đồ quan hệ



Hình 2.1.1: Lược đồ thiết kế cơ sở dữ liệu quản lý vé xem phim

1NF Tất cả các bảng hình 2.1.1 chứa thuộc tính đơn trị (theo yêu cầu kinh doanh).

Bản ghi trong mỗi bảng đều phân biệt (Xem thêm mục 2.3). Cơ sở dữ liệu ở dạng chuẩn thứ nhất.

2NF Cơ sở dữ liệu ở 1NF. Tất cả các thuộc tính phi khóa (non-key) phụ thuộc hàm (functionally dependent) vào toàn bộ khóa chính. Không xảy ra hiện tượng phụ thuộc một phần. Cơ sở dữ liệu ở dạng chuẩn thứ hai.

3NF & BCNF Cơ sở dữ liệu ở 2NF. Tất cả phụ thuộc hàm trong các quan hệ đều có dạng $X \rightarrow Y$ với X là siêu khóa. Cơ sở dữ liệu ở dạng chuẩn Boyce–Codd, do đó cũng ở dạng chuẩn thứ ba.

2.2 Ràng buộc

2.2.1 Ràng buộc miền

DC1: Giá chỗ ngồi chiếm tối đa 20% giá niêm yết của bộ phim và tối thiểu bằng giá niêm yết, hệ số giá chỗ ngồi nằm từ 1 đến 1.2

$$\forall t \in \text{seat_fees}, \quad 1 \leq t.\text{FeeFactor} \leq 1.20.$$

DC2: Lương của nhân viên bán vé tối thiểu là 5 triệu đồng

$$\forall t \in \text{sellors}, \quad t.\text{Salary} \geq 5000000.$$

DC3: Thời lượng của một bộ phim, về căn bản, phải tối thiểu 1 phút.

$$\forall t \in \text{films}, \quad t.\text{Duration} \geq 00 : 01 : 00.$$

DC4: Giá phim niêm yết, về căn bản, phải tối thiểu 1 nghìn đồng.

$$\forall t \in \text{films}, \quad t.\text{Price} \geq 1000.$$

2.2.2 Ràng buộc bảng

KC: Mọi khóa chính K trong tất cả các quan hệ R phải là Unique:

$$\forall t_1, t_2 \in R, \quad \text{if } t_1 \neq t_2 \text{ then } t_1.K \neq t_2.K. \text{ Xem mục 2.3.}$$

UC1: Cặp $(\text{SlotID}, \text{PlayDate})$ trong quan hệ play_slots phải là Unique, vì mỗi ngày chỉ chiếu một bộ phim ở một suất chiếu nhất định. (Đồng thời đảm bảo ở BCNF.)

UC2: Cặp $(\text{BookID}, \text{SeatID})$ trong quan hệ booking_seats phải là Unique.

UC3: Cặp $(\text{SeatArea}, \text{SeatNumber})$ trong seats là Unique

Các ràng buộc Unique khác được trình bày trong mục 2.3.

2.2.3 Ràng buộc tham chiếu

Thêm: Xem hình 2.1.1

Sửa: Mọi chỉnh sửa đều là cascade¹.

Xóa: Các quy tắc xóa được trình bày trực quan và rõ ràng trong hình 2.1.1 và mục 2.3

2.3 Thông tin các bảng

SEATS						
Field	Data type	PK	FK	Unique	Nullable	On delete
SeatID	char(3)	✓		✓		
SeatArea	char(1)		✓			Restrict
SeatNumber	varchar(2)					

SEAT_FEES						
Field	Data type	PK	FK	Unique	Nullable	On delete
SeatArea	char(1)	✓		✓		
FeeFactor	numeric(3, 2)					

CUSTOMERS						
Field	Data type	PK	FK	Unique	Nullable	On delete
CustID	char(12)	✓		✓		
CustFirstName	varchar(64)				✓	
CustLastName	varchar(32)				✓	
CustPhone	varchar(16)			✓		
Hash	char(60)					
Salt	char(22)					

¹“Database is not for everyone. If you don’t know what cascade or restrict is, you’re better off doing something else.” – Nhật Bui.

SELLERS

Field	Data type	PK	FK	Unique	Nullable	On delete
SellerID	char(8)	✓		✓		
SellerFirstName	varchar(64)					
SellerLastName	varchar(32)					
Salary	integer					
SellerPhone	varchar(16)			✓		
SellerAddress	varchar(64)					

BOOKING_DETAILS

Field	Data type	PK	FK	Unique	Nullable	On delete
BookID	char(16)	✓		✓		
PlaySlotID	char(12)		✓			Restrict
TheaterID	char(3)		✓			Restrict
BookTime	timestamp					

BOOKINGS

Field	Data type	PK	FK	Unique	Nullable	On delete
BookID	char(16)		✓	✓		Cascade
CustID	char(12)		✓			Cascade

BOOKING_SEATS

Field	Data type	PK	FK	Unique	Nullable	On delete
BookID	char(16)		✓			Cascade
SeatID	char(3)		✓			Restrict

BOOKING_SELLERS

Field	Data type	PK	FK	Unique	Nullable	On delete
BookID	char(16)		✓	✓		Cascade
SellerID	char(8)		✓			Cascade

THEATERS

Field	Data type	PK	FK	Unique	Nullable	On delete
TheaterID	char(3)	✓		✓		
TheaterCity	varchar(32)			✓		

GENRES

Field	Data type	PK	FK	Unique	Nullable	On delete
GenreID	char(2)	✓		✓		
Genre	varchar(32)			✓		

FILMS

Field	Data type	PK	FK	Unique	Nullable	On delete
FilmID	char(8)	✓		✓		
FilmName	varchar(128)					
GenreID	char(2)		✓			Restrict
Director	varchar(64)				✓	
ReleaseDate	date				✓	
Duration	interval					
Price	integer					

SLOTS

Field	Data type	PK	FK	Unique	Nullable	On delete
SlotID	char(3)	✓		✓		
SlotStartTime	time			✓		

PLAY_SLOTS

Field	Data type	PK	FK	Unique	Nullable	On delete
PlaySlotID	char(12)	✓		✓		
FilmID	char(8)		✓			Restrict
SlotID	char(3)		✓			Restrict
PlayDate	date					

2.4 Truy vấn bằng đại số quan hệ

- 1 Cho biết các bộ phim thuộc thể loại *Comedy*.

$$\Pi_{\text{FilmID}, \text{FilmName}, \text{Genre}, \text{Director}}(\sigma_{\text{Genre} = 'Comedy'}(\text{films} \bowtie \text{genres})).$$

- 2 Cho biết các bộ phim được chiếu trong tháng 11.

$$\Pi_{\text{FilmID}, \text{FilmName}, \text{PlayDate}}(\sigma_{\text{PlayDate} \geq '2021-11-01' \wedge \text{PlayDate} \leq '2021-11-30'}(\text{play_slots} \bowtie \text{films})).$$

- 5 Cho biết các phim chiếu trong khung giờ 20h - 23h (phân biệt với suất chiếu).

$$\Pi_{\text{FilmID}, \text{FilmName}}(\sigma_{\text{SlotStartTime} \leq 23:00:00 \wedge \text{SlotStartTime} + \text{Duration} \geq 20:00:00}(\text{play_slots} \bowtie \text{slots} \bowtie \text{films})).$$

- 7 Cho biết số lượng vé được bán tại từng rạp.

$$\Pi_{\text{TheaterID}} \gamma_{\text{count}(\text{SeatID})}(\text{booking_seats} \bowtie \text{theaters})$$

Các yêu cầu truy vấn khác hoặc phức tạp, hoặc yêu cầu subquery trong select hoặc where nên không thể viết một cách trực tiếp dưới dạng đại số quan hệ.

Cài đặt cơ sở dữ liệu

3.1 Định nghĩa dữ liệu

Tạo cơ sở dữ liệu

```
create database film_tickets
with
  owner = nhat
  encoding = 'UTF8'
connection limit = -1;
```

Tạo bảng

```
create table seats (
  SeatID char(3) primary key,
  SeatArea char(1) not null,
  SeatNumber varchar(2) not null,
  unique(SeatArea, SeatNumber) -- UC3
)
create table seat_fees (
  SeatArea char(1) primary key,
  FeeFactor numeric(3, 2) not null -- DC1
  constraint chk_factor check (FeeFactor >= 1 and FeeFactor <= 1.2)
)
create table customers (
  CustID char(12) primary key,
  CustFirstName varchar(64),
  CustLastName varchar(32),
  CustPhone varchar(16) unique,
  Hash char(60) not null,
  Salt char(22) not null
)
create table sellers (
  SellerID char(8) primary key,
  SellerFirstName varchar(64) not null,
  SellerLastName varchar(32) not null,
```

```
Salary int not null check (Salary >= 5000000), -- DC2
SellerPhone varchar(16) not null unique,
SellerAddress varchar(64) not null
)
create table booking_details (
    BookID char(16) primary key,
    PlaySlotID char(12) not null,
    TheaterID char(3) not null,
    BookTime timestamp not null
)
create table bookings (
    BookID char(16) not null unique,
    CustID char(12) not null
)
create table booking_sellers (
    BookID char(16) not null unique,
    SellerID char(8) not null
)
create table booking_seats (
    BookID char(16) not null,
    SeatID char(3) not null,
    unique(BookID, SeatID) -- UC2
)
create table theaters (
    TheaterID char(3) primary key,
    TheaterCity varchar(32) not null unique
)
create table genres (
    GenreID char(2) primary key,
    Genre varchar(32) not null unique
)
create table films (
    FilmID char(8) primary key,
    FilmName varchar(128) not null,
    GenreID char(2) not null,
    Director varchar(64),
    ReleaseDate date,
    Duration interval not null check (Duration >= '00:01:00'), -- DC3
    Price int not null check (Price >= 1000) -- DC4
)
create table slots (
    SlotID char(3) primary key,
    SlotStartTime time not null unique
```

```
)  
create table play_slots (  
    PlaySlotID char(12) primary key,  
    FilmID char(8) not null,  
    SlotID char(3) not null,  
    PlayDate date not null,  
    unique(SlotID, PlayDate) -- UC1  
)
```

Tạo tham chiếu

```
alter table seats  
    add constraint fk_seat_area  
        foreign key (SeatArea) references seat_fees  
        on delete restrict  
  
alter table booking_details  
    add constraint fk_play_slot  
        foreign key (PlaySlotID) references play_slots  
        on delete restrict,  
    add constraint fk_theater  
        foreign key (TheaterID) references theaters  
        on delete restrict  
  
alter table bookings  
    add constraint fk_booking_detail  
        foreign key (BookID) references booking_details  
        on delete cascade,  
    add constraint fk_customer  
        foreign key (CustID) references customers  
        on delete cascade  
  
alter table booking_seats  
    add constraint fk_booking_detail  
        foreign key (BookID) references booking_details  
        on delete cascade,  
    add constraint fk_seat  
        foreign key (SeatID) references seats  
        on delete cascade  
  
alter table booking_sellers  
    add constraint fk_booking_detail  
        foreign key (BookID) references booking_details  
        on delete cascade,
```

```

add constraint fk_seller
foreign key (SellerID) references sellers
on delete cascade

alter table films
add constraint fk_genre
foreign key (GenreID) references genres
on delete restrict

alter table play_slots
add constraint fk_film
foreign key (FilmID) references films
on delete restrict,
add constraint fk_slot
foreign key (SlotID) references slots
on delete restrict

```

3.2 Thêm dữ liệu

Thêm dữ liệu vào bảng seat_fees

```

insert into seat_fees values
('A', 1.2), ('B', 1.15), ('C', 1.1), ('D', 1.05), ('E', 1.0)

```

	seatarea [PK] character (1)	feefactor numeric (3,2)
1	A	1.20
2	B	1.15
3	C	1.10
4	D	1.05
5	E	1.00

Từ đây trở đi, dữ liệu sẽ được thêm bằng cách bulk loading từ các tệp csv chuẩn bị sẵn, thay vì dùng 100 câu lệnh insert into. Ảnh chụp **chỉ** bao gồm vài **dòng** đầu tiên của bảng.

```

copy seats
from '/home/nhatbui/Documents/University/CSDL/data/seats.csv'
delimiter ',' csv header;

```

	seatid [PK] character (3)	seatarea character (1)	seatnumber character varying (2)
1	ztd	A	10
2	fwn	E	20
3	1jm	E	10
4	5ug	D	10
5	wy8	C	10

copy genres

from '/home/nhatbui/Documents/University/CSDL/data/genres.csv'
 delimiter ',' csv header;

	genreid [PK] character (2)	genre character varying (32)
1	11	Action & adventure
2	12	Animation
3	15	Comedy
4	17	Documentary
5	18	Drama

copy films

from '/home/nhatbui/Documents/University/CSDL/data/films.csv'
 delimiter ',' csv header;

	filmid [PK] character (8)	filmname character varying (128)	genreid character (2)	director character varying (64)	releasedate date	duration interval	price integer
1	ZZ1DOSQB	Godzilla vs. Kong	11	Adam Wingard	2021-01-01	01:53:00	120000
2	QS7QLYIH	Harry Potter and the Chambe...	11	Chris Columbus	2002-01-01	02:40:00	100000
3	J2G3UNWN	Spectre	11	Sam Mendes	2015-01-01	02:28:00	110000
4	GBLB4UMG	Spider-Man: Far From Home	11	Jon Watts	2019-01-01	02:09:00	115000
5	MZ299HHO	The Nightmare Before Christ...	12	Henry Selick	1993-01-01	01:16:00	75000
6	S851ARI0	Raya and the Last Dragon	12	Don Hall	2021-01-01	01:47:00	100000
7	VVQMDA4P	Cruella	15	Craig Gillespie	2021-01-01	02:13:00	100000

copy theaters

from '/home/nhatbui/Documents/University/CSDL/data/theaters.csv'
 delimiter ',' csv header;

	theaterid [PK] character (3)	theatercity character varying (32)
1	AAA	Hà Nội
2	LAK	Mã Lai
3	OOP	TP.HCM

copy slots

from '/home/nhatbui/Documents/University/CSDL/data/slots.csv'
 delimiter ',' csv header;

	slotid [PK] character (3)	slotstarttime time without time zone
1	789	08:00:00
2	785	09:00:00
3	781	10:00:00
4	777	11:00:00

```
copy customers
from '/home/nhatbui/Documents/University/CSDL/data/customers.csv'
delimiter ',' csv header;
```

	custid [PK] character (12)	custfirstname character varying	custlastname character varying	custphone character varying	hash character (60)	salt character (22)
1	GZL0LwifRbQ	Erik	Jefferson	0123456789	\$2y\$10\$eLULm.Wmvin...	Fv5buwDKj1IJkWuPJF...
2	IFcJCSKogOo7	John Fuck	Kennedy	0985413508	\$2y\$10\$DDb5/PB0K/q...	Felo4fK3kycMXMQvab0...
3	9381lbSTPEkw	Huê	Cây	089741505...	\$2y\$10\$1e.69KYgHdD...	LvyTi1bzsGEXRqHdJEH...
4	oBx4wMXoxVjL	Bùi Bất	Bại	016969696...	\$2y\$10\$ImI2SA/zRJof...	ihVKxdQE0qqGbHG5Td...
5	Nztq2doYUcug	Chu Chích	Chèo	047552236...	\$2y\$10\$wWhHY8VVhx...	HIHUIPJUTGgP59WD4...
6	gclb7pWwbAJ0	Luôn Vui	Tươi	025845484...	\$2y\$10\$0j2Xi255qWm...	FctUixdyBBjYehJZ5cap...
7	bw5g2IS2yE1G	Chà	Neo	099985554...	\$2y\$10\$RUqbh2nl9JO...	YD9N3Q1MHX1Ye52N...
8	UnlfKO9qmX7k	Đi	Ó	047774576...	\$2y\$10\$MD.8O6uDfXv...	FyRtmdVBzi9oaOLU0Lg...
9	Mh2iETsyjqKF	A Léch Xang Đờ	Quang	098765431...	\$2y\$10\$9XbxRly/Z2DO...	7JzLa0K3MvkSsAVm6...
10	rdVBQ5zUtWGv	Rô Nan	Đồ	177777777...	\$2y\$10\$9XbxRly/Z2DO...	HGN05cng5pWd049uXf...

```
copy sellers
from '/home/nhatbui/Documents/University/CSDL/data/sellers.csv'
delimiter ',' csv header;
```

	sellerid [PK] character (8)	sellerfirstname character varying	sellerlastname character varying	salary integer	sellerphone character varying (15)	selleraddress character varying (64)
1	RGDVRgGL	Tony	Tèo	10000000	0454545454545	911 Hách Cơ, Quận 0, Mã Lai
2	UkL7L5vM	A Nô Ny	Một	8500000	0674646547115	72 Bạch Đằng, Tân Bình, TP.HCM
3	ac7bHbP8	Đôn	Chè	6900000	0644872454545	169 Hai Bà Trưng, Quận 3, TP.HCM
4	qjPwzmxU	Giogious	Armani	5100000	0988711054748	123 Giải Phóng, Hoàn Kiếm, Hà Nội
5	IFuDlnAJ	Cường Đồ	Na	8000000	0556432132654	3 Tân Kỳ Tân Quý, Ba Đình, Hà Nội

```
copy play_slots
from '/home/nhatbui/Documents/University/CSDL/data/play_slots.csv'
delimiter ',' csv header;
```

	playslotid [PK] character (12)	filmid character (8)	slotid character (3)	playdate date
1	aasTSjzyohna	MZ299HHO	785	2021-10-31
2	iLtwpFcQ55PK	N7L1YJCK	753	2021-10-31
3	Tg8MfQtNBfRA	BBWU1AMG	777	2021-11-05
4	MhgD8T0kJHcj	JPPKTb3N	741	2021-11-05
5	MbORiFsk00B8	MZ299HHO	745	2021-11-06

```
copy booking_details
from '/home/nhatbui/Documents/University/CSDL/data/booking_details.csv'
delimiter ',' csv header;
```

	bookid [PK] character (16)	playslotid character (12)	theaterid character (3)	booktime timestamp without time zone
1	3yjYKVF5SPq27zh2	iLtwpFcQ55PK	LAK	2021-10-25 20:50:00
2	57GD62mo5CDUCjvs	aasTSjzyohna	LAK	2021-10-31 08:06:00
3	5Vjxx81vK1QpaPa6	aasTSjzyohna	AAA	2021-10-31 08:27:00
4	6f3IYAvXDPKk8PLK	iLtwpFcQ55PK	AAA	2021-10-31 16:40:00
5	AHOarvtunpB3thD	iLtwpFcQ55PK	OOP	2021-10-31 16:25:00
6	aIJvZGxrSUNPv14k	aasTSjzyohna	LAK	2021-10-31 08:05:00
7	aQDVg6FpYInpbj87	aasTSjzyohna	OOP	2021-10-31 08:50:00
8	atddVlqn6N4D7tTM	MbORiFsk00B8	OOP	2021-11-01 09:31:00
9	bto1xTPc6LiORmUp	MbORiFsk00B8	LAK	2021-11-01 11:11:00
10	C3WFh0QsIfv59kWD	aasTSjzyohna	LAK	2021-10-31 08:24:00

```
copy bookings
from '/home/nhatbui/Documents/University/CSDL/data/bookings.csv'
delimiter ',' csv header;
```

	bookid character (16)	custid character (12)
1	phEBGU5mzMuuN2U6	Mh2iETsyjqKF
2	t3LUPShCMzelPGmZ	gcIb7pWwbAJ0
3	6f3IYAvXDPKk8PLK	GZL0LwiflRbQ
4	XcjFrja4pKkPwbly	oBx4wMXoxVjL
5	PgyakiZ8r1GRRzMG	Mh2iETsyjqKF
6	uPg5GWZygECk5Hli	oBx4wMXoxVjL
7	aIJvZGxrSUNPv14k	GZL0LwiflRbQ
8	57GD62mo5CDUCjvs	bw5g2IS2yE1G

```
copy booking_sellers
from '/home/nhatbui/Documents/University/CSDL/data/booking_sellers.csv'
delimiter ',' csv header;
```

	bookid character (16)	sellerid character (8)
1	57GD62mo5CDUCjvs	RGDVDRgL
2	5Vjxx81vK1QpaPa6	qjPwzmxU
3	6f3IYAvXDPKk8PLK	IFuDlnAJ
4	AHOarvtunpB3thD	ac7bHbP8
5	aIJvZGxrSUNPv14k	RGDVDRgL
6	aQDVg6FpYInpbj87	ac7bHbP8
7	C3WFh0QsIfv59kWD	RGDVDRgL

```
copy booking_seats
```

```
from '/home/nhatbui/Documents/University/CSDL/data/booking_seats.csv'
delimiter ',' csv header;
```

	bookid character (16)	seatid character (3)
1	phEBGU5mzMuuN2U6	9vu
2	5Vjxx81vK1QpaPa6	pox
3	t3LUPShCMzeIPGmZ	p21
4	6f3iYAvXDPKk8PLK	h0j
5	j6GNePZ7UjPOeKL5	fqq
6	tl1m4RRHSj47luJp	vrz
7	tl1m4RRHSj47luJp	waw
8	tl1m4RRHSj47luJp	9pp

3.3 Truy vấn bằng SQL

1. Các bộ phim thuộc thể loại *Comedy*.

```
select FilmID, FilmName, Genre, Director
from films natural join genres
where genre = 'Comedy'
```

	filmid character (8)	filmname character varying (128)	genre character varying (32)	director character varying (64)
1	VVQMDA4P	Cruella	Comedy	Craig Gillespie
2	3HMOBRSG	Beetlejuice	Comedy	Tim Burton

2. Các bộ phim được chiếu trong tháng 11.

```
select FilmID, FilmName, Director, PlayDate
from films natural join play_slots
where PlayDate >= '2021-11-01'
and PlayDate <= '2021-11-30'
```

	filmid character (8)	filmname character varying (128)	director character varying (64)	playdate date
1	MZ299HHO	The Nightmare Before Christmas	Henry Selick	2021-11-06
2	BBWU1AMG	The Night House	David Bruckner	2021-11-05
3	JPPKT83N	Interstellar	Christopher Nolan	2021-11-05

3. Tỷ lệ đặt vé trực tuyến

```

select (count(*) - (select count(*)
                    from booking_sellers)
       ) / cast(count(*) as numeric)
       * 100 as percent
from booking_details;

```

	percent numeric
1	20.00000000

4. Những khách hàng đi xem nhiều hơn một lần trong tháng 11.

```

select CustID,
       CustFirstName + ' ' + CustLastName as CustomerName,
       count(*) as Times
from   customers natural join bookings
       natural join booking_details
       natural join play_slots
where  PlayDate between '2021-11-01' and '2021-11-30'
group by CustID, CustomerName
having count(*) > 1

```

	custid [PK] character (12)	customername text	times bigint
1	GZL0LwiflRbQ	Erik Jefferson	2
2	Mh2iETsyjqKF	A Lếch Xang Đờ Quang	2
3	oBx4wMXoxVjL	Bùi Bất Bại	4

5. các phim chiếu vào khung giờ 20h - 23h (phân biệt với suất chiếu).

```

select FilmID,
       FilmName,
       SlotStartTime as StartTime,
       SlotStartTime + Duration as EndTime
from   films natural join play_slots
       natural join slots
where  SlotStartTime <= '23:00:00'
       and SlotStartTime + Duration >= '20:00:00'

```

	filmid character (8)	filmname character varying (128)	starttime time without t	endtime time without t
1	JPPKT83N	Interstellar	20:00:00	22:49:00
2	MZ299HHO	The Nightmare Before Christmas	19:00:00	20:16:00

6. Tính giá hóa đơn có mã là *tl1m4RRHSj47luJp*

```
with film_price as (
    select Price
    from
        booking_details natural join play_slots
        natural join films
    where BookID = 'tl1m4RRHSj47luJp'
),
seat_fee as (
    select FeeFactor
    from
        booking_details natural join booking_seats
        natural join seats
        natural join seat_fees
    where BookID = 'tl1m4RRHSj47luJp'
)
select sum(Price * FeeFactor) as BillPrice
from film_price, seat_fee;
```

	billprice numeric
1	262500.00

7. Cho biết số lượng vé bán tại từng rạp

```
select TheaterID, TheaterCity, count(*) as Solds
from booking_details natural join theaters
group by TheaterID, TheaterCity;
```

	theaterid character (3)	theatercity character varying (32)	solds bigint
1	OOP	TP.HCM	8
2	AAA	Hà Nội	10
3	LAK	Mã Lai	12

8. Cho biết tổng doanh thu toàn bộ rạp.

```

with film_price as (
  select BookID, Price
  from
    booking_details natural join play_slots
    natural join films
), seat_fee as (
  select BookID, FeeFactor
  from
    booking_details natural join booking_seats
    natural join seats
    natural join seat_fees
)
select sum(Price * FeeFactor) as Revenue
from film_price join seat_fee using (BookID);

```

	revenue numeric
1	5759500.00

9. Cho biết phim mang lại doanh thu nhiều nhất.

```

with film_revenues as (
  select
    FilmID,
    FilmName,
    sum(Price * FeeFactor) as revenue
  from
    booking_details natural join play_slots
    natural join films
    natural join booking_seats
    natural join seats
    natural join seat_fees
  group by FilmID, FilmName
)
select FilmID, FilmName, Revenue
from film_revenues
where Revenue = (select max(Revenue)
                 from film_revenues)

```

	filmid character (8)	filmname character varying (128)	revenue numeric
1	BBWU1AMG	The Night House	1979250.00

10. Cho biết nhân viên bán được nhiều vé nhất.

```
select
  SellerID,
  SellerFirstName + ' ' + SellerLastName as SellerName
from
  sellers natural join booking_sellers
group by SellerID
having count(BookID) >= all(select count(*)
                             from booking_sellers
                             group by SellerID)
```

	sellerid [PK] character (8)	sellername text
1	RGDVDRgL	Tony Tèo

Kết luận

Tiểu luận về cơ bản đã đặc tả bài toán quản lý vé xem phim, và thực hiện các yêu cầu truy vấn¹ của doanh nghiệp. Cơ sở dữ liệu đã được thiết kế tỉ mỉ để giảm thiểu trùng lặp dữ liệu. Phần ràng buộc và thông tin bảng được mô tả đủ để một lập trình viên có thể cài đặt chính xác. Vì thời gian có hạn, tiểu luận không bao gồm hai chi tiết: rạp có nhiều màn chiếu, và phim có hai định dạng 2D/3D.

¹“Nobody cares if the query is simple or complicated. They only care whether it solves a real-world problem or not.” – Nhat Bui.