



รายงานฐานข้อมูลระบบจองตั๋วหนัง  
MOVIE TICKET BOOKING SYSTEM DATABASE REPORT

กลุ่ม DB2SS5

นายสุทธิพงษ์ จงจำรัสพันธ์	66130500085
นายอรรถกร ผิวสุข	66130500087
นายพงศธร สามกลิ่น	66130500118
นายภวินท์ นาเจริญ	66130500121
นายภัทรวุฒิ ชัยเดช	66130500122

งานนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

วิทยาศาสตร์บัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ

คณะเทคโนโลยีสารสนเทศ

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี

ปีการศึกษา 2567

### สารบัญ

การมีส่วนร่วมของสมาชิกในทีม .....	3
Business Requirement .....	4
Logical Database Design.....	6
SQL Statement.....	7
Data Dictionary.....	12
DDL Script for creating the database .....	19
DDL Script for insert data to table.....	24

## การมีส่วนร่วมของสมาชิกในทีม

### นายสุทธิพงษ์ จงจำรัสพันธ์

- มีส่วนร่วมในงาน 20%
- รับผิดชอบ:
  - ออกแบบ Logical Database Design (ร่วมกับ นายภวินท์ นาเจริญ)
  - มีส่วนร่วมในการเขียน Business Requirements
  - ตรวจสอบ และ ปรับแก้เนื้อหาต่าง ๆ ที่เพื่อนช่วยกันทำ

### นายอรรถกร ผิวสุข

- มีส่วนร่วมในงาน 20%
- รับผิดชอบ:
  - เขียนตัวอย่างคำสั่ง SQL Statements
  - มีส่วนร่วมในการเขียน Business Requirements

### นายพงศธร สามกลิ่น

- มีส่วนร่วมในงาน 20%
- รับผิดชอบ:
  - จัดทำ Data Dictionary
  - มีส่วนร่วมในการเขียน Business Requirements

### นายภวินท์ นาเจริญ

- มีส่วนร่วมในงาน 20%
- รับผิดชอบ:
  - ออกแบบ Logical Database Design (ร่วมกับ นายสุทธิพงษ์ จงจำรัสพันธ์)

- เขียน DDL Script สำหรับสร้างฐานข้อมูล
- มีส่วนร่วมในการเขียน Business Requirements

#### นายภัทรวุฒิ ชัยเดช

- มีส่วนร่วมในงาน 20%
- รับผิดชอบ:
  - เขียน DDL Script สำหรับเพิ่มข้อมูลจำลอง (Mockup Data)
  - มีส่วนร่วมในการเขียน Business Requirements

## Business Requirement

### 1. โรงภาพยนตร์และสาขา

- ระบบต้องสามารถจัดเก็บข้อมูลของโรงภาพยนตร์หลายสาขาได้ โดยแต่ละสาขามีข้อมูล เช่น ชื่อสาขา ที่อยู่ จังหวัด และเบอร์โทรศัพท์
- ในแต่ละสาขาสถาบันมีหลายโรงฉายภาพยนตร์ (Theatre) ได้ ซึ่งแต่ละโรงมีหมายเลขเฉพาะ จำนวนที่นั่ง และเชื่อมโยงกับสาขานั้น

### 2. ข้อมูลโรงภาพยนตร์และการฉายภาพยนตร์

- โรงภาพยนตร์แต่ละโรงสามารถมีรอบฉายภาพยนตร์ได้หลายรอบ แต่ละรอบการฉายจะมีข้อมูล เช่น วันที่และเวลาที่ฉาย ภาษาที่ฉาย คำบรรยาย และโรงที่ฉาย
- ภาพยนตร์แต่ละเรื่องสามารถฉายได้ในหลายรอบและหลายโรง โดยแต่ละเรื่องจะมีข้อมูลชื่อภาพยนตร์ ระยะเวลา วันออกฉาย เรตติ้ง และคำอธิบาย

### 3. ลูกค้าและการจองตั๋ว

- ระบบต้องจัดเก็บข้อมูลของลูกค้า เช่น ชื่อลูกค้า อีเมล เบอร์โทรศัพท์ และที่อยู่
- ลูกค้าสามารถจองตั๋วได้หลายครั้ง โดยแต่ละการจองจะบันทึกวันที่ทำการจอง รอบการฉายที่จอง สถานะของการจอง และข้อมูลการชำระเงินหากมี

#### 4. ที่นั่งและประเภทที่นั่ง

- แต่ละโรงภาพยนตร์มีที่นั่งหลายที่ ซึ่งแต่ละที่นั่งมีหมายเลขเฉพาะ
- ที่นั่งแต่ละที่จะมีประเภทที่นั่ง เช่น ธรรมดา อำนวยาน ซึ่งระบบต้องจัดเก็บข้อมูลประเภทที่นั่งนั้นๆไว้

โดยแต่ละประเภทมีราคาไม่เท่ากัน

#### 5. การชำระเงิน

- แต่ละการจองตัวสามารถชำระเงินได้หลายวิธี เช่น เงินสด บัตรเครดิต

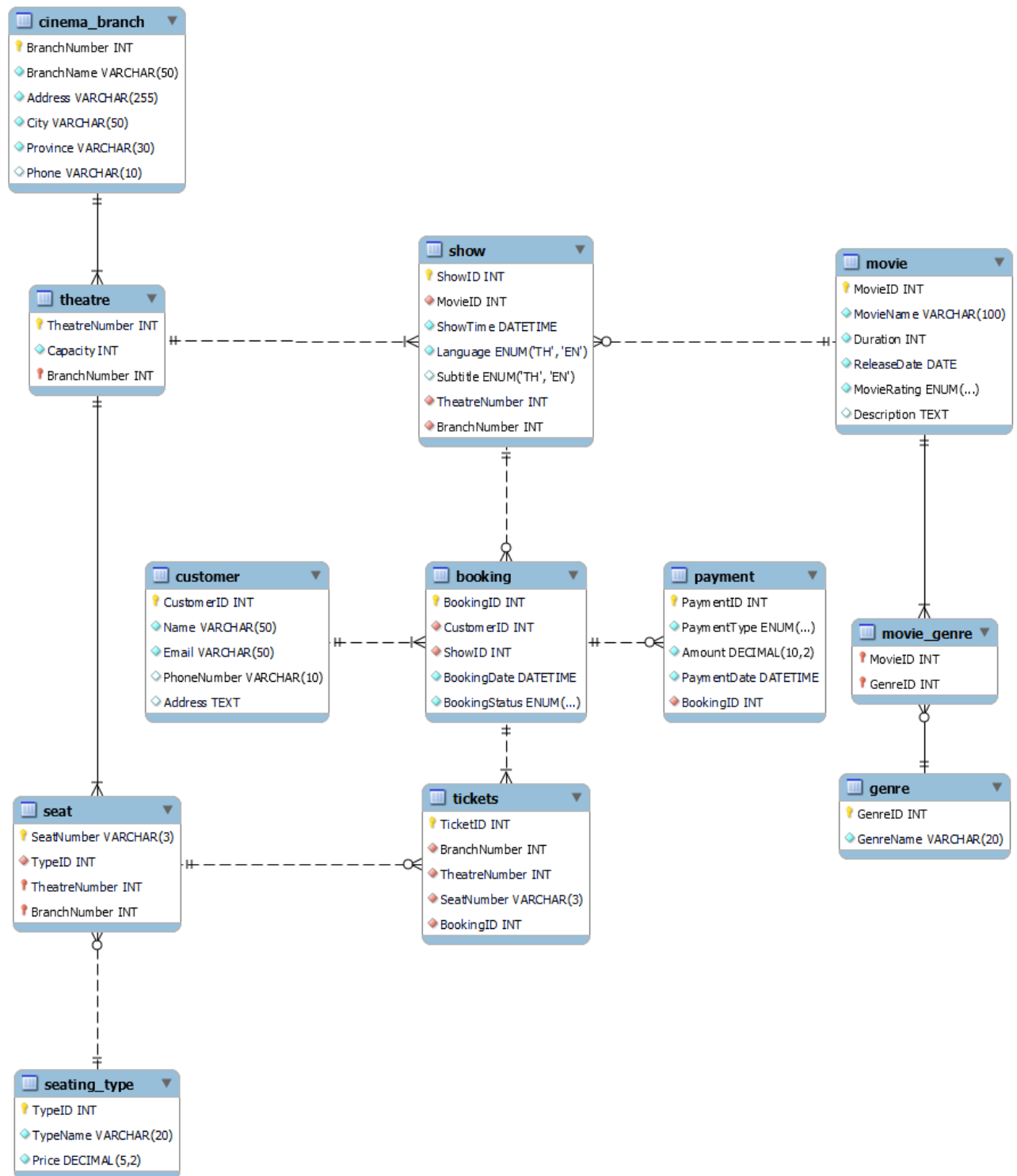
โดยแต่ละรายการชำระเงินจะมีจำนวนเงิน วันที่และเวลาชำระ และเชื่อมโยงกับการจองที่เกี่ยวข้อง  
ทั้งนี้ลูกค้าสามารถจองไว้ก่อนได้โดยไม่จำเป็นต้องจ่ายเงินในทันที

#### 6. ประเภทภาพยนตร์

- ระบบต้องสามารถจัดเก็บข้อมูลประเภทของภาพยนตร์ เช่น แอ็คชั่น ตลก

โดยภาพยนตร์แต่ละเรื่องสามารถมีหลายประเภท

## Logical Database Design



## SQL Statement

1. แสดงรายละเอียดของแต่ละสาขาของโรงภาพยนตร์ รวมถึงชื่อสาขา ที่อยู่ เมือง จังหวัด หมายเลขโรงภาพยนตร์ ความจุ และเบอร์โทรศัพท์ โดยเรียงตามหมายเลขสาขา

```
select cb.branchname , cb.address , cb.city , cb.province
, t.TheatreNumber , t.capacity , cb.phone
from cinema_branch cb
join theatre t on cb.branchnumber = t.branchnumber
order by cb.branchnumber;
```

	branchname	address	city	province	TheatreNumber	capacity	phone
▶	Downtown Cinema	123 Main St	Bangkok Yai	Bangkok	1	150	0201234567
	Downtown Cinema	123 Main St	Bangkok Yai	Bangkok	2	200	0201234567
	Riverside Mall Cinema	456 River Rd	Muang Chiang Mai	Chiang Mai	1	180	0891234567
	Riverside Mall Cinema	456 River Rd	Muang Chiang Mai	Chiang Mai	2	120	0891234567
	Beachside Cinema	789 Beach Blvd	Kathu	Phuket	1	160	0641234567
	Beachside Cinema	789 Beach Blvd	Kathu	Phuket	2	140	0641234567
	Suburban Cineplex	321 Green Ave	Pak Kret	Nonthaburi	1	220	0321234567
	Suburban Cineplex	321 Green Ave	Pak Kret	Nonthaburi	2	180	0321234567
	City Center Cinema	654 Urban St	Pathum Wan	Bangkok	1	190	0741234567

2. ดึงข้อมูลภาพยนตร์ทั้งหมดพร้อมรายละเอียด รวมถึงชื่อภาพยนตร์ ประเภท คำอธิบาย ความยาวการฉาย เรตติ้ง วันที่ฉาย เวลาในการฉาย ภาษา คำบรรยาย และหมายเลขโรงภาพยนตร์ โดยเรียงตามชื่อภาพยนตร์.

```
select m.MovieName , g.GenreName , m.Description , m.Duration
, m.MovieRating , m.ReleaseDate , sh.ShowTime
, sh.Language , sh.Subtitle , sh.TheatreNumber
from movie m join moviebookingsystem.show sh on m.movieID = sh.movieID
join movie_genre mn on m.movieid = mn.movieid
join genre g on mn.genreid = g.genreid
order by m.movienam;
```

	MovieName	GenreName	Description	Duration	MovieRating	ReleaseDate	ShowTime	Language	Subtitle	TheatreNumber
▶	Cyber Heist	Action	A thrilling heist movie set in a digital world.	125	General	2023-07-20	2023-12-04 18:00:00	EN	TH	1
	Digital Frontier	Sci-Fi	A sci-fi thriller that explores the future of techn...	145	13+	2023-05-20	2023-12-03 19:30:00	EN	TH	1
	Heart of the City	Drama	A heartwarming drama about love and friendshi...	110	General	2023-06-15	2023-12-02 18:00:00	EN	TH	1
	Heart of the City	Drama	A heartwarming drama about love and friendshi...	110	General	2023-06-15	2023-12-02 20:00:00	TH	TH	2
	Love in Bangkok	Romance	A romantic story set in the heart of Bangkok.	90	General	2023-02-14	2023-12-01 21:00:00	TH	EN	2
	Quantum Strike	Action	An action-packed thriller that keeps you on the ...	120	18+	2023-01-01	2023-12-01 19:00:00	EN	TH	1
	Urban Legends	Horror	A horror film that brings urban legends to life.	100	18+	2023-10-31	2023-12-03 21:30:00	TH	EN	2
	Urban Legends	Horror	A horror film that brings urban legends to life.	100	18+	2023-10-31	2023-12-04 20:00:00	TH	TH	2
	Urban Legends	Horror	A horror film that brings urban legends to life.	100	18+	2023-10-31	2023-12-05 21:00:00	TH	TH	2
	Whispers of Time	Drama	A magical tale that transcends time and space.	115	General	2023-09-30	2023-12-05 19:00:00	EN	EN	1

3. แสดงข้อมูลลูกค้าและการจองของพวกเขา เช่น ชื่อลูกค้า อีเมล เบอร์โทรศัพท์ ที่อยู่ หมายเลขการจอง ชื่อภาพยนตร์ วันที่จอง และสถานะการจอง โดยเรียงตามชื่อลูกค้า

```
select c.name , c.email , c.phonenumber , c.address , b.bookingid
, m.moviename , b.bookingdate , b.bookingstatus
from customer c join booking b on c.customerid = b.customerid
join moviebookingsystem.show sh on b.showid = sh.showid
join movie m on sh.movieID = m.movieID
order by c.name;
```

name	email	phonenumber	address	bookingid	moviename	bookingdate	bookingstatus
Anupong Thongchai	anupong@gmail.com	0856789012	654 Ramkhamhaeng Road, Bangkok, Thailand	6	Digital Frontier	2024-11-18 01:28:43	PAID
Anupong Thongchai	anupong@gmail.com	0856789012	654 Ramkhamhaeng Road, Bangkok, Thailand	11	Urban Legends	2024-11-04 03:23:45	PAID
Malee Wattana	malee@gmail.com	0823456789	456 Silom Road, Bangkok, Thailand	3	Love in Bangkok	2024-11-09 14:24:42	CANCELLED
Malee Wattana	malee@gmail.com	0823456789	456 Silom Road, Bangkok, Thailand	8	Cyber Heist	2024-11-01 11:01:11	CANCELLED
Prasert Charoenkij	prasert@gmail.com	0834567890	789 Ratchadaphisek Road, Bangkok, Thailand	4	Heart of the City	2024-11-09 16:35:08	CANCELLED
Prasert Charoenkij	prasert@gmail.com	0834567890	789 Ratchadaphisek Road, Bangkok, Thailand	9	Urban Legends	2024-11-25 19:07:43	PAID
Somchai Srisai	somchai@gmail.com	0812345678	123 Sukhumvit Road, Bangkok, Thailand	1	Quantum Strike	2024-11-08 09:17:19	PAID
Somchai Srisai	somchai@gmail.com	0812345678	123 Sukhumvit Road, Bangkok, Thailand	2	Love in Bangkok	2024-12-05 10:12:59	PAID
Somchai Srisai	somchai@gmail.com	0812345678	123 Sukhumvit Road, Bangkok, Thailand	7	Urban Legends	2024-11-11 12:16:46	PENDING
Suchada Poomee	suchada@gmail.com	0845678901	321 Lat Phrao Road, Bangkok, Thailand	5	Heart of the City	2024-11-10 11:54:53	PAID
Suchada Poomee	suchada@gmail.com	0845678901	321 Lat Phrao Road, Bangkok, Thailand	10	Whispers of Time	2024-11-14 22:55:23	PENDING

4. ดึงข้อมูลที่นั่งในโรงภาพยนตร์แต่ละสาขา รวมถึงชื่อสาขา หมายเลขโรงภาพยนตร์ หมายเลขที่นั่ง ประเภทที่นั่ง และราคาของที่นั่ง โดยเรียงตามหมายเลขสาขา หมายเลขโรง และหมายเลขที่นั่ง

```
select cb.branchname , s.theatrenumber , s.seatnumber , st.typename , st.price from cinema_branch cb
join theatre t on cb.branchnumber = t.branchnumber
join seat s on t.branchnumber = s.branchnumber
join seating_type st on s.typeid = st.typeid
order by s.branchnumber , s.theatrenumber , s.seatnumber ;
```



branchname	theatrenumber	seatnumber	typename	price
Downtown Cinema	1	A1	Regular	169.00
Downtown Cinema	1	A1	Regular	169.00
Downtown Cinema	1	A2	Regular	169.00
Downtown Cinema	1	A2	Regular	169.00
Downtown Cinema	1	A3	Regular	169.00
Downtown Cinema	1	A3	Regular	169.00
Downtown Cinema	1	A4	Regular	169.00
Downtown Cinema	1	A4	Regular	169.00
Downtown Cinema	1	B1	Deluxe	249.00
Downtown Cinema	1	B1	Deluxe	249.00
Downtown Cinema	1	B2	Deluxe	249.00
Downtown Cinema	1	B2	Deluxe	249.00

5. แสดงข้อมูลการชำระเงินของลูกค้าสำหรับการจองแต่ละครั้ง เช่น หมายเลขการชำระเงิน ชื่อลูกค้า

ประเภทการชำระเงิน จำนวนเงิน วันที่ชำระ และหมายเลขการจอง โดยเรียงตามหมายเลขการชำระเงิน

```
select p.paymentid , c.name , p.paymenttype , p.amount , p.paymentdate , b.bookingid from payment p
join booking b on p.bookingid = b.bookingid join customer c on b.customerid = c.customerid
order by p.paymentid;
```

paymentid	name	paymenttype	amount	paymentdate	bookingid
1	Somchai Srisai	Credit Card	338.00	2024-11-20 03:23:45	1
2	Somchai Srisai	Debit card	699.00	2024-11-09 03:23:45	2
3	Malee Wattana	cash	169.00	2024-11-11 03:23:45	3
4	Prasert Charoenkij	PromptPay	418.00	2024-11-15 03:23:45	4
5	Suchada Poomee	Credit Card	249.00	2024-11-08 03:23:45	5
6	Anupong Thongchai	Debit card	699.00	2024-11-03 03:23:45	6
7	Somchai Srisai	cash	169.00	2024-11-07 03:23:45	7
8	Malee Wattana	PromptPay	418.00	2024-12-01 03:23:45	8

6. ค้นหาข้อมูลลูกค้าและที่นั่งที่พวกเขาจองไว้ รวมถึงชื่อลูกค้า หมายเลขตั๋ว ชื่อภาพยนตร์ หมายเลขที่นั่ง ประเภทที่นั่ง และวันที่จอง โดยเรียงตามชื่อลูกค้าและหมายเลขที่นั่ง

```

select distinct c.name, t.TicketID , m.moviename , s.SeatNumber , st.TypeName , b.bookingdate
from customer c
join booking b on c.customerid = b.customerid
join moviebookingsystem.show sh on b.showid = sh.showid
join movie m on sh.movieID = m.movieID
join tickets t on b.bookingid = t.Bookingid
join seat s on t.SeatNumber = s.SeatNumber
join seating_type st on s.TypeID = st.TypeID
order by c.name , s.seatnumber;

```

name	TicketID	moviename	SeatNumber	TypeName	bookingdate
Anupong Thongchai	11	Digital Frontier	B1	Deluxe	2024-11-18 01:28:43
Anupong Thongchai	12	Digital Frontier	B2	Deluxe	2024-11-18 01:28:43
Malee Wattana	5	Love in Bangkok	A1	Regular	2024-11-09 14:24:42
Malee Wattana	6	Love in Bangkok	A2	Regular	2024-11-09 14:24:42
Malee Wattana	15	Cyber Heist	C1	VIP	2024-11-01 11:01:11
Malee Wattana	16	Cyber Heist	C2	VIP	2024-11-01 11:01:11
Prasert Charoenkij	7	Heart of the City	C1	VIP	2024-11-09 16:35:08
Prasert Charoenkij	8	Heart of the City	C2	VIP	2024-11-09 16:35:08
Somchai Srisai	1	Quantum Strike	A1	Regular	2024-11-08 09:17:19
Somchai Srisai	13	Urban Legends	A1	Regular	2024-11-11 12:16:46
Somchai Srisai	2	Quantum Strike	A2	Regular	2024-11-08 09:17:19
Somchai Srisai	14	Urban Legends	A2	Regular	2024-11-11 12:16:46

7. นับจำนวนการจองของลูกค้าแต่ละคน และจัดกลุ่มตามชื่อลูกค้า โดยเรียงตามชื่อในลำดับตัวอักษร

```

select c.name, count(c.name) as number_of_booking from customer c
join booking b on c.customerid = b.customerid
group by c.name
order by c.name;

```

name	number_of_booking
Anupong Thongchai	2
Malee Wattana	2
Prasert Charoenkij	2
Somchai Srisai	3
Suchada Poomee	2

8. นับจำนวนภาพยนตร์ในแต่ละประเภท และจัดกลุ่มตามชื่อประเภท

โดยเรียงตามชื่อประเภทในลำดับตัวอักษร

```
select g.genrename , count(g.genrename) as count_of_gerne from genre g
join movie_genre mg on g.genreid = mg.genreid
group by g.genrename
order by g.genrename;
```

genrename	count_of_gerne
Action	2
Adventure	1
Drama	2
Horror	2
Romance	1
Sci-Fi	1
Thriller	1

9. ค้นหาข้อมูลตั๋วหนังของลูกค้าที่ชื่อ Malee Wattana โดยมี เลขตั๋ว ชื่อหนัง เลขที่นั่ง ประเภทที่นั่ง และรอบฉาย

```
select distinct t.TicketID , m.moviename , s.SeatNumber , st.TypeName , sh.ShowTime
from customer c
join booking b on c.customerid = b.customerid
join moviebookingsystem.show sh on b.showid = sh.showid
join movie m on sh.movieID = m.movieID
join tickets t on b.bookingid = t.Bookingid
join seat s on t.SeatNumber = s.SeatNumber
join seating_type st on s.TypeID = st.TypeID
where c.name = "Malee Wattana" ;
```

TicketID	moviename	SeatNumber	TypeName	ShowTime
6	Love in Bangkok	A2	Regular	2023-12-01 21:00:00
5	Love in Bangkok	A1	Regular	2023-12-01 21:00:00
16	Cyber Heist	C2	VIP	2023-12-04 18:00:00
15	Cyber Heist	C1	VIP	2023-12-04 18:00:00

10.แก้ไขสถานะการจองของ เลขการจองที่ 3 ให้เป็น cancelled

```
update booking  
set bookingstatus = "CANCELLED"  
where bookingid = 3;
```

BookingID	CustomerID	ShowID	BookingDate	BookingStatus
1	1	1	2024-11-08 09:17:19	PAID
2	1	2	2024-12-05 10:12:59	PAID
3	2	2	2024-11-09 14:24:42	CANCELLED

## Data Dictionary

Table: Cinema\_Branch

Description: ข้อมูลสาขา

No	Column Name	Description	Data Type	Constraint	Referenced Table
1	BranchNumber	รหัสสาขา	INT	Primary Key, Not Null	
2	BranchName	ชื่อสาขา	VARCHAR(50)	Not Null	
3	Address	ที่อยู่	VARCHAR(255)	Not Null	
4	City	เมือง	VARCHAR(50)	Not Null	
5	Province	จังหวัด	VARCHAR(30)	Not Null	
6	phone	เบอร์โทร	VARCHAR(10)		

Table: Theater

Description: โรงหนัง

No	Column Name	Description	Data Type	Constraint	Referenced Table
1	TheatreNumber	รหัสโรงหนัง	INT	Primary Key, Not Null	
2	Capacity	ความจุ	INT	Not Null	
3	BranchNumber	รหัสสาขา	INT	Foreign Key, Not Null	cinema_branch

Table: Show

Description: ข้อมูลรอบฉาย

No	Column Name	Description	Data Type	Constraint	Referenced Table
1	ShowID	รหัสรอบฉาย	INT	Primary Key, Not Null, Auto Increment	
2	MovieID	รหัสหนัง	INT	Foreign Key, Not Null	movie
3	ShowTime	เวลาที่ฉาย	DATETIME	Not Null	
4	Language	ภาษา	ENUM('TH','EN')	Not Null	
5	Subtitle	คำแปล	ENUM('TH','EN','')		
6	TheatreNumber	รหัสโรงหนัง	INT	Foreign Key, Not Null	theatre
7	BranchNumber	รหัสสาขา	INT	Foreign Key, Not Null	cinema_branch

Table: Movie

Description: ข้อมูลหนัง

No	Column Name	Description	Data Type	Constraint	Referenced Table
1	MovieID	รหัสหนัง	INT	Primary Key, Not Null, Auto Increment	
2	MovieName	ชื่อหนัง	VARCHAR(100)	Not Null	
3	Duration	ระยะเวลา	INT	Not Null	
4	ReleaseDate	วันที่ออกฉาย	DATE	Not Null	
5	MovieRating	คะแนนเรตติ้ง	ENUM('General', '13+', '15+', '18+', '20+')	Not Null	
6	Description	รายละเอียด	TEXT		

Table: Movie\_Genre

Description: ประเภทของหนังที่เชื่อมกับหนัง

No	Column Name	Description	Data Type	Constraint	Referenced Table
1	MovieID	รหัสหนัง	INT	Foreign Key, Not Null	movie
2	GenreID	รหัสประเภทหนัง	INT	Foreign Key, Not Null	genre

Table: Genre

Description: ประเภทของหนัง

No	Column Name	Description	Data Type	Constraint	Referenced Table
1	GenreID	รหัสประเภทของหนัง	INT	Primary Key, Not Null	
2	GenreName	ชื่อประเภทของหนัง	VARCHAR(20)	Not Null	

Table: Customer

Description: ข้อมูลของลูกค้า

No	Column Name	Description	Data Type	Constraint	Referenced Table
1	CustomerID	รหัสลูกค้า	INT	Primary Key, Not Null, Auto Increment	
2	Name	ชื่อลูกค้า	VARCHAR(50)	Not Null	
3	Email	อีเมล	VARCHAR(50)	Not Null,	
4	PhoneNumber	เลขเบอร์โทร	VARCHAR(10)		
5	Address	ที่อยู่	TEXT		

Table: Payment

Description: ข้อมูลการจ่ายเงิน



No	Column Name	Description	Data Type	Constraint	Referenced Table
1	PaymentID	รหัสการจ่ายเงิน	INT	Primary Key, Not Null, Auto Increment	
2	PaymentType	ประเภทการชำระเงิน	ENUM('Credit Card ', 'PromptPay', 'Debit Card','Cash')	Not Null	
3	Amount	จำนวนเงิน	DECIMAL(10,2)	Not Null	
4	PaymentDate	วันที่จ่ายเงิน	DATETIME	Not Null	
5	BookingID	รหัสการจอง	INT	Foreign Key, Not Null, Unique	booking

Table: Booking

Description: ข้อมูลของลูกค้า

No	Column Name	Description	Data Type	Constraint	Referenced Table
1	BookingID	รหัสการจอง	INT	Primary Key, Not Null, Auto Increment	
2	CustomerID	รหัสลูกค้า	INT	Foreign Key, Not Null	customer
3	ShowID	รหัสรอบฉาย	INT	Foreign Key, Not Null	show
4	BookingDate	วันที่จอง	DATETIME	Not Null	
5	BookingStatus	สถานะการจอง	ENUM('PENDING', 'PAID', 'CANCELLED')	Not Null	

Table: Seat

Description: ข้อมูลที่นั่ง

No	Column Name	Description	Data Type	Constraint	Referenced Table
1	SeatNumber	รหัสที่นั่ง	VARCHAR(3)	Primary Key, Not Null	
2	TypeID	รหัสประเภทที่นั่ง	INT	Foreign Key, Not Null	seating_type
3	TheatreNumber	รหัสโรงหนัง	INT	Foreign Key, Not Null	theatre
4	BranchNumber	รหัสสาขา	INT	Foreign Key, Not Null	cinema_branch

Table: Seating\_Type

Description: ข้อมูลประเภทของที่นั่ง

No	Column Name	Description	Data Type	Constraint	Referenced Table
1	TypeID	รหัสประเภทที่นั่ง	INT	Primary Key, Not Null	
2	TypeName	ชื่อประเภทที่นั่ง	VARCHAR(20)	Not Null	
3	Price	ราคา	DECIMAL(5,2)	Not Null	

Table: Booking\_Seat

Description: ข้อมูลการจองที่เชื่อมกับที่นั่ง

No	Column Name	Description	Data Type	Constraint	Referenced Table
1	TicketID	รหัสตั๋ว	INT	Primary Key, Not Null, Auto Increment	
2	BookingID	รหัสการจอง	INT	Foreign Key, Not Null	booking
3	SeatNumber	เลขที่นั่ง	VARCHAR(3)	Foreign Key, Not Null	seat
4	TheatreNumber	รหัสโรงหนัง	INT	Foreign Key, Not Null	theatre
5	BranchNumber	รหัสสาขา	INT	Foreign Key, Not Null	cinema_branch

## DDL Script for creating the database

```
-- MySQL Workbench Forward Engineering SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0; SET
@OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0; SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_S
UBSTITUTION';

-- -----
-- Schema mydb
-- -----
-- Schema moviebookingsystem
-- -----
DROP SCHEMA IF EXISTS `moviebookingsystem` ;

-- -----
-- Schema moviebookingsystem
-- -----
CREATE SCHEMA IF NOT EXISTS `moviebookingsystem` DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci ;
USE `moviebookingsystem` ;

-- -----
-- Table `customer`
```

```

-----
DROP TABLE IF EXISTS `customer` ;
CREATE TABLE IF NOT EXISTS `customer` (
  `CustomerID` INT NOT NULL AUTO_INCREMENT,
  `Name` VARCHAR(50) NOT NULL,
  `Email` VARCHAR(50) NOT NULL,
  `PhoneNumber` VARCHAR(10) NULL DEFAULT NULL,
  `Address` TEXT NULL DEFAULT NULL,
  PRIMARY KEY (`CustomerID`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;
-----

-- Table `cinema_branch`
-----

DROP TABLE IF EXISTS `cinema_branch` ;
CREATE TABLE IF NOT EXISTS `cinema_branch` (
  `BranchNumber` INT NOT NULL,
  `BranchName` VARCHAR(50) NOT NULL,
  `Address` VARCHAR(255) NOT NULL,
  `City` VARCHAR(50) NOT NULL,
  `Province` VARCHAR(30) NOT NULL,
  `Phone` VARCHAR(10) NULL DEFAULT NULL,
  PRIMARY KEY (`BranchNumber`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;
-----

-- Table `theatre`
-----

DROP TABLE IF EXISTS `theatre` ;
CREATE TABLE IF NOT EXISTS `theatre` (
  `TheatreNumber` INT NOT NULL,
  `Capacity` INT NOT NULL,
  `BranchNumber` INT NOT NULL,
  PRIMARY KEY (`TheatreNumber`, `BranchNumber`),
  INDEX `fk_theatre_cinema_branch1_idx` (`BranchNumber` ASC) VISIBLE,
  CONSTRAINT `fk_theatre_cinema_branch1`
  FOREIGN KEY (`BranchNumber`)
  REFERENCES `cinema_branch` (`BranchNumber`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;
-----

-- Table `movie`
-----

DROP TABLE IF EXISTS `movie` ;
CREATE TABLE IF NOT EXISTS `movie` (
  `MovieID` INT NOT NULL AUTO_INCREMENT,
  `MovieName` VARCHAR(100) NOT NULL,
  `Duration` INT NOT NULL,

```

```

`ReleaseDate` DATE NOT NULL,
`MovieRating` ENUM('General', '13+', '15+', '18+', '20+') NOT NULL,
`Description` TEXT NULL DEFAULT NULL,
PRIMARY KEY (`MovieID`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;

-----

-- Table `show`
-----

DROP TABLE IF EXISTS `show` ;
CREATE TABLE IF NOT EXISTS `show` (
`ShowID` INT NOT NULL AUTO_INCREMENT,
`MovieID` INT NOT NULL,
`ShowTime` DATETIME NOT NULL,
`Language` ENUM('TH', 'EN') NOT NULL,
`Subtitle` ENUM('TH', 'EN') NULL DEFAULT NULL,
`TheatreNumber` INT NOT NULL,
`BranchNumber` INT NOT NULL,
PRIMARY KEY (`ShowID`),
INDEX `Movie_ID` (`MovieID` ASC) VISIBLE,
INDEX `fk_show_theatre1_idx` (`TheatreNumber` ASC, `BranchNumber` ASC) VISIBLE,
CONSTRAINT `fk_show_theatre1`
FOREIGN KEY (`TheatreNumber` , `BranchNumber`)
REFERENCES `theatre` (`TheatreNumber` , `BranchNumber`),
CONSTRAINT `show_ibfk_1`
FOREIGN KEY (`MovieID`)
REFERENCES `movie` (`MovieID`)
ON DELETE CASCADE)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;

-----

-- Table `booking`
-----

DROP TABLE IF EXISTS `booking` ;
CREATE TABLE IF NOT EXISTS `booking` (
`BookingID` INT NOT NULL AUTO_INCREMENT,
`CustomerID` INT NOT NULL,
`ShowID` INT NOT NULL,
`BookingDate` DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
`BookingStatus` ENUM('PENDING', 'PAID', 'CANCELLED') NOT NULL,
PRIMARY KEY (`BookingID`),
INDEX `Customer_ID` (`CustomerID` ASC) VISIBLE,
INDEX `Show_ID` (`ShowID` ASC) VISIBLE,
CONSTRAINT `booking_ibfk_1` FOREIGN KEY (`CustomerID`)
REFERENCES `customer` (`CustomerID`)
ON DELETE CASCADE,
CONSTRAINT `booking_ibfk_2`
FOREIGN KEY (`ShowID`)

```

```

REFERENCES `show` (`ShowID`)
ON DELETE CASCADE)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;

-----
-- Table `genre`
-----

DROP TABLE IF EXISTS `genre` ;
CREATE TABLE IF NOT EXISTS `genre` (
  `GenreID` INT NOT NULL,
  `GenreName` VARCHAR(20) NOT NULL,
  PRIMARY KEY (`GenreID`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;

-----
-- Table `movie_genre`
-----

DROP TABLE IF EXISTS `movie_genre` ;
CREATE TABLE IF NOT EXISTS `movie_genre` (
  `MovieID` INT NOT NULL,
  `GenreID` INT NOT NULL,
  PRIMARY KEY (`MovieID`,
  `GenreID`),
  INDEX `fk_Genre_has_movie_movie1_idx` (`MovieID` ASC) VISIBLE,
  INDEX `fk_Genre_has_movie_Genre1_idx` (`GenreID` ASC) VISIBLE,
  CONSTRAINT `fk_Genre_has_movie_Genre1`
  FOREIGN KEY (`GenreID`)
  REFERENCES `genre` (`GenreID`),
  CONSTRAINT `fk_Genre_has_movie_movie1`
  FOREIGN KEY (`MovieID`)
  REFERENCES `movie` (`MovieID`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;

-----
-- Table `payment`
-----

DROP TABLE IF EXISTS `payment` ;
CREATE TABLE IF NOT EXISTS `payment` (
  `PaymentID` INT NOT NULL AUTO_INCREMENT,
  `PaymentType` ENUM('Credit Card', 'PromptPay', 'Debit Card', 'Cash') NOT NULL,
  `Amount` DECIMAL(10,2) NOT NULL,
  `PaymentDate` DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `BookingID` INT NOT NULL,
  PRIMARY KEY (`PaymentID`),
  INDEX `fk_payment_booking1_idx` (`BookingID` ASC) VISIBLE,
  UNIQUE INDEX `BookingID_UNIQUE` (`BookingID` ASC) VISIBLE,
  CONSTRAINT `fk_payment_booking1`

```

```

FOREIGN KEY ('BookingID')
REFERENCES 'booking' ('BookingID'))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;

-----

-- Table 'seating_type'
-----

DROP TABLE IF EXISTS 'seating_type' ;
CREATE TABLE IF NOT EXISTS 'seating_type' (
  'TypeID' INT NOT NULL,
  'TypeName' VARCHAR(20) NOT NULL,
  'Price' DECIMAL(5,2) NOT NULL,
  PRIMARY KEY ('TypeID'))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;

-----

-- Table 'seat'
-----

DROP TABLE IF EXISTS 'seat' ;
CREATE TABLE IF NOT EXISTS 'seat' (
  'SeatNumber' VARCHAR(3) NOT NULL,
  'TypeID' INT NOT NULL,
  'TheatreNumber' INT NOT NULL,
  'BranchNumber' INT NOT NULL,
  PRIMARY KEY ('SeatNumber',
  'TheatreNumber', 'BranchNumber'),
  INDEX 'fk_seat_type1_idx' ('TypeID' ASC) VISIBLE,
  INDEX 'fk_seat_theatre1_idx' ('TheatreNumber' ASC, 'BranchNumber' ASC) VISIBLE,
  CONSTRAINT 'fk_seat_theatre1'
  FOREIGN KEY ('TheatreNumber' , 'BranchNumber')
  REFERENCES 'theatre' ('TheatreNumber' , 'BranchNumber'),
  CONSTRAINT 'fk_seat_type1'
  FOREIGN KEY ('TypeID')
  REFERENCES 'seating_type' ('TypeID'))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;

-----

-- Table 'tickets'
-----

DROP TABLE IF EXISTS 'tickets' ;
CREATE TABLE IF NOT EXISTS 'tickets' (
  'TicketID' INT NOT NULL AUTO_INCREMENT,
  'BranchNumber' INT NOT NULL,
  'TheatreNumber' INT NOT NULL,
  'SeatNumber' VARCHAR(3) NOT NULL,
  'BookingID' INT NOT NULL,
  PRIMARY KEY ('TicketID'),

```

```

INDEX `fk_seat_has_booking_booking1_idx` (`BookingID` ASC) VISIBLE,
INDEX `fk_seat_has_booking_seat1_idx` (`SeatNumber` ASC, `TheatreNumber` ASC, `BranchNumber` ASC) VISIBLE,
CONSTRAINT `fk_seat_has_booking_seat1`
FOREIGN KEY (`SeatNumber` , `TheatreNumber` , `BranchNumber`)
REFERENCES `seat` (`SeatNumber` , `TheatreNumber` , `BranchNumber`)
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT `fk_seat_has_booking_booking1`
FOREIGN KEY (`BookingID`)
REFERENCES `booking` (`BookingID`)
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;
SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;

```

## DDL Script for insert data to table

use moviebookingsystem;

```

SET FOREIGN_KEY_CHECKS = 0;
INSERT INTO customer (Name, Email, PhoneNumber, Address)
VALUES
('Somchai Srisai', 'somchai@gmail.com', '0812345678', '123 Sukhumvit Road, Bangkok, Thailand'),
('Malee Wattana', 'malee@gmail.com', '0823456789', '456 Silom Road, Bangkok, Thailand'),
('Prasert Charoenkij', 'prasert@gmail.com', '0834567890', '789 Ratchadaphisek Road, Bangkok, Thailand'),
('Suchada Poomee', 'suchada@gmail.com', '0845678901', '321 Lat Phrao Road, Bangkok, Thailand'),
('Anupong Thongchai', 'anupong@gmail.com', '0856789012', '654 Ramkhamhaeng Road, Bangkok, Thailand');
INSERT INTO cinema_branch (BranchNumber, BranchName, Address, City, Province, Phone)
VALUES
(1, 'Downtown Cinema', '123 Main St', 'Bangkok Yai', 'Bangkok', '0201234567'),
(2, 'Riverside Mall Cinema', '456 River Rd', 'Muang Chiang Mai', 'Chiang Mai', '0891234567'),
(3, 'Beachside Cinema', '789 Beach Blvd', 'Kathu', 'Phuket', '0641234567'),
(4, 'Suburban Cineplex', '321 Green Ave', 'Pak Kret', 'Nonthaburi', '0321234567'),
(5, 'City Center Cinema', '654 Urban St', 'Pathum Wan', 'Bangkok', '0741234567');
INSERT INTO theatre (TheatreNumber, Capacity, BranchNumber)
VALUES
(1, 150, 1),
(2, 200, 1),
(1, 180, 2),
(2, 120, 2),
(1, 160, 3),
(2, 140, 3),
(1, 220, 4),
(2, 180, 4),
(1, 190, 5);
INSERT INTO movie (MovieName, Duration, ReleaseDate, MovieRating, Description)

```



## VALUES

(Quantum Strike', 120, '2023-01-01', '18+', 'An action-packed thriller that keeps you on the edge of your seat.'),  
 (Love in Bangkok', 90, '2023-02-14', 'General', 'A romantic story set in the heart of Bangkok.'),  
 (Midnight Shadows', 105, '2023-03-01', '18+', 'A chilling horror film that will leave you breathless.'),  
 (Expedition Unknown', 130, '2023-04-10', 'General', 'An adventurous journey through uncharted territories.'),  
 (Digital Frontier', 145, '2023-05-20', '13+', 'A sci-fi thriller that explores the future of technology.'),  
 (Heart of the City', 110, '2023-06-15', 'General', 'A heartwarming drama about love and friendship in urban life.'),  
 (Cyber Heist', 125, '2023-07-20', 'General', 'A thrilling heist movie set in a digital world.'),  
 (Royal Escape', 140, '2023-08-25', '18+', 'A royal family on the run in an epic adventure.'),  
 (Whispers of Time', 115, '2023-09-30', 'General', 'A magical tale that transcends time and space.'),  
 (Urban Legends', 100, '2023-10-31', '18+', 'A horror film that brings urban legends to life.');

INSERT INTO 'show' (MovieID, ShowTime, Language, Subtitle, TheatreNumber, BranchNumber)

## VALUES

(1, '2023-12-01 19:00:00', 'EN', 'TH', 1, 1), -- Theatre 1 in Branch 1  
 (2, '2023-12-01 21:00:00', 'TH', 'EN', 2, 1), -- Theatre 2 in Branch 1  
 (6, '2023-12-02 18:00:00', 'EN', NULL, 1, 2), -- Theatre 1 in Branch 2  
 (6, '2023-12-02 20:00:00', 'TH', 'TH', 2, 2), -- Theatre 2 in Branch 2  
 (5, '2023-12-03 19:30:00', 'EN', 'TH', 1, 3), -- Theatre 1 in Branch 3  
 (10, '2023-12-03 21:30:00', 'TH', 'EN', 2, 3), -- Theatre 2 in Branch 3  
 (7, '2023-12-04 18:00:00', 'EN', 'TH', 1, 4), -- Theatre 1 in Branch 4  
 (10, '2023-12-04 20:00:00', 'TH', NULL, 2, 4), -- Theatre 2 in Branch 4  
 (9, '2023-12-05 19:00:00', 'EN', 'EN', 1, 5), -- Theatre 1 in Branch 5  
 (10, '2023-12-05 21:00:00', 'TH', 'TH', 2, 5); -- Theatre 2 in Branch 5

INSERT INTO booking (CustomerID, ShowID, BookingDate, BookingStatus) VALUES

(1, 1, '2024-11-08 09:17:19', 'PAID'),  
 (1, 2, '2024-12-05 10:12:59', 'PAID'),  
 (2, 2, '2024-11-09 14:24:42', 'PENDING'),  
 (3, 3, '2024-11-09 16:35:08', 'CANCELLED'),  
 (4, 4, '2024-11-10 11:54:53', 'PAID'),  
 (5, 5, '2024-11-18 01:28:43', 'PAID'),  
 (1, 6, '2024-11-11 12:16:46', 'PENDING'),  
 (2, 7, '2024-11-01 11:01:11', 'CANCELLED'),  
 (3, 8, '2024-11-25 19:07:43', 'PAID'),  
 (4, 9, '2024-11-14 22:55:23', 'PENDING'),  
 (5, 10, '2024-11-04 03:23:45', 'PAID');

INSERT INTO genre (GenreID, GenreName)

## VALUES

(1, 'Action'),  
 (2, 'Romance'),  
 (3, 'Horror'),  
 (4, 'Adventure'),  
 (5, 'Sci-Fi'),  
 (6, 'Drama'),  
 (7, 'Thriller'),  
 (8, 'Comedy');

INSERT INTO movie\_genre (MovieID, GenreID)

## VALUES

(1, 1), -- Quantum Strike - Action  
 (2, 2), -- Love in Bangkok - Romance  
 (3, 3), -- Midnight Shadows - Horror

(4, 4), -- Expedition Unknown - Adventure

(5, 5), -- Digital Frontier - Sci-Fi

(6, 6), -- Heart of the City - Drama

(7, 1), -- Cyber Heist - Action

(8, 7), -- Royal Escape - Thriller

(9, 6), -- Whispers of Time - Drama

(10, 3); -- Urban Legends - Horror

INSERT INTO seating\_type (TypeID, TypeName, Price)

VALUES

(1, 'Regular', 169.00),

(2, 'Deluxe', 249.00),

(3, 'VIP', 699.00);

INSERT INTO seat (SeatNumber, TypeID, TheatreNumber, BranchNumber) VALUES

('A1', 1, 1, 1), ('A2', 1, 1, 1), ('A3', 1, 1, 1), ('A4', 1, 1, 1), ('B1', 2, 1, 1), ('B2', 2, 1, 1), ('B3', 2, 1, 1), ('B4', 2, 1, 1), ('C1', 3, 1, 1), ('C2', 3, 1, 1), ('C3', 3, 1, 1), ('C4', 3, 1, 1), ('A1', 1, 2, 1), ('A2', 1, 2, 1), ('A3', 1, 2, 1), ('A4', 1, 2, 1), ('B1', 2, 2, 1), ('B2', 2, 2, 1), ('B3', 2, 2, 1), ('B4', 2, 2, 1), ('C1', 3, 2, 1), ('C2', 3, 2, 1), ('C3', 3, 2, 1), ('C4', 3, 2, 1);

INSERT INTO tickets (BranchNumber, TheatreNumber, SeatNumber, BookingID)

VALUES

(1, 1, 'A1', 1),

(1, 1, 'A2', 1),

(1, 1, 'B1', 2),

(1, 1, 'B2', 2),

(1, 2, 'A1', 3),

(1, 2, 'A2', 3),

(1, 2, 'C1', 4),

(1, 2, 'C2', 4),

(2, 1, 'A1', 5),

(2, 1, 'A2', 5),

(2, 1, 'B1', 6),

(2, 1, 'B2', 6),

(2, 2, 'A1', 7),

(2, 2, 'A2', 7),

(2, 2, 'C1', 8),

(2, 2, 'C2', 8);

INSERT INTO payment (PaymentType, Amount, PaymentDate, BookingID) VALUES

('Credit Card', 338.00, '2024-11-20 03:23:45', 1),

('Debit Card', 699.00, '2024-11-09 03:23:45', 2),

('Cash', 169.00, '2024-11-11 03:23:45', 3),

('PromptPay', 418.00, '2024-11-15 03:23:45', 4),

('Credit Card', 249.00, '2024-11-08 03:23:45', 5),

('Debit Card', 699.00, '2024-11-03 03:23:45', 6),

('Cash', 169.00, '2024-11-07 03:23:45', 7),

('PromptPay', 418.00, '2024-12-01 03:23:45', 8);

SET FOREIGN\_KEY\_CHECKS = 1;