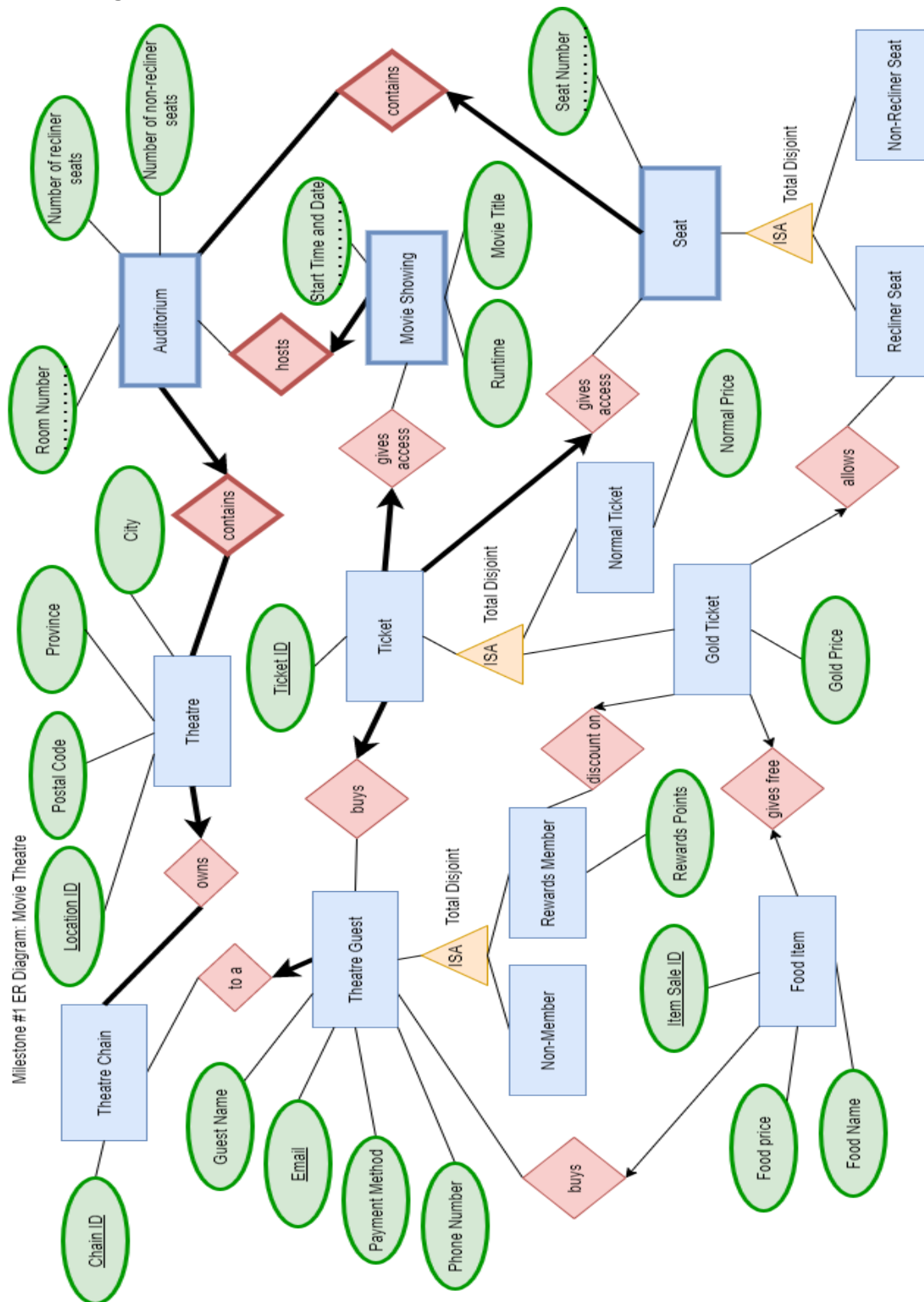


2) ER Diagram



Change in ER Model: Many-to-one relationship between Gold Ticket and Rewards Member.

3) Relation Model

1. NonMember
 - a. NonMember(Email: Char(20), Name: Char(20), PayMethod: Char(20), PhoneNum: Char(11), ChainID: Char(20))
 - b. Specifications
 - i. Primary Key: Email
 - ii. Candidate Key(s): Email, PhoneNum
 - iii. Foreign Keys: ChainID
 - iv. Other Constraints:
 1. TheaterChain NOT NULL
 2. PhoneNum UNIQUE
2. RewMember
 - a. RewMember(Email: Char(20), Name: Char(20), PayMethod: Char(20), PhoneNum: Char(11), ChainID: Char(20), RewPoints: Integer)
 - b. Specifications
 - i. Primary Key: Email
 - ii. Candidate Key(s): Email, PhoneNum
 - iii. Foreign Keys: ChainID
 - iv. Other Constraints:
 1. TheaterChain NOT NULL
 2. PhoneNum UNIQUE
3. FoodItem
 - a. FoodItem(ItemSaleID: Integer, Price: Integer, Name: Integer, MemberEmail: Char(20), TicketID: Integer)
 - b. Specifications
 - i. Primary Key: ItemSaleID
 - ii. Candidate Key(s): ItemSaleID
 - iii. Foreign Keys: MemberEmail, TicketID
 - iv. Other Constraints:
 1. TicketID UNIQUE
4. NonRecSeat
 - a. NonRecSeat(SeatNum: Integer, RoomNum: Integer, LocID: Integer)
 - b. Specifications
 - i. Primary Key: SeatNum, RoomNum, LocID
 - ii. Candidate Key(s): {SeatNum, RoomNum, LocID}
 - iii. Foreign Keys: RoomNum, LocID
 - iv. Other Constraints: none

5. RecSeat

- a. RecSeat(SeatNum: Integer, RoomNum: Integer, LocID: Integer)
- b. Specifications
 - i. Primary Key: SeatNum, RoomNum, LocID
 - ii. Candidate Key(s): {SeatNum, RoomNum, LocID}
 - iii. Foreign Keys: RoomNum, LocID
 - iv. Other Constraints: none

6. Auditorium

- a. Auditorium(RoomNum: Integer, LocID: Integer, NumRecSeats: Integer, NumNonRecSeats: Integer)
- b. Specifications
 - i. Primary Key: RoomNum, LocID
 - ii. Candidate Key(s): none
 - iii. Foreign Keys: LocID
 - iv. Other Constraints:
 - 1. NumRecSeats NOT NULL
 - 2. NumNonRecSeats NOT NULL

7. Theater

- a. Theater(LocID: Integer, PostalCode: Char(6), Province: Char(20), City: Char(20), ChainID: Integer)
- b. Specifications
 - i. Primary Key: LocID
 - ii. Candidate Key(s): LocID
 - iii. Foreign Keys: ChainID
 - iv. Other Constraints:
 - 1. ChainID NOT NULL

8. TheaterChain

- a. TheaterChain(ChainID: Integer)
- b. Specifications
 - i. Primary Key: ChainID
 - ii. Candidate Key(s): ChainID
 - iii. Foreign Keys: none
 - iv. Other Constraints: none

9. NormalTicket

- a. NormalTicket(TicketID: Integer, NPrice: Integer, STimeDate: Char(20), RoomNum: Integer, LocID: Integer, SeatNum: Integer, Email: Char(20))
- b. Specifications
 - i. Primary Key: TicketID
 - ii. Candidate Key(s): TicketID
 - iii. Foreign Keys: Email, STimeDate, RoomNum, LocID, SeatNum
 - iv. Other Constraints:

University of British Columbia, Vancouver

Department of Computer Science

1. Email NOT NULL
2. STimeDate NOT NULL
3. RoomNum NOT NULL
4. LocID NOT NULL
5. SeatNum NOT NULL

10. GoldTicket

- a. GoldTicket(TicketID: Integer, GPrice: Integer, STimeDate: Char(20),
RoomNum: Integer, LocID: Integer, SeatNum: Integer,
Email: Char(20), ItemSaleID: Integer)
- b. Specifications
 - i. Primary Key: TicketID
 - ii. Candidate Key(s): TicketID
 - iii. Foreign Keys: STimeDate, RoomNum, LocID, SeatNum, ItemSaleID
 - iv. Other Constraints:
 1. Email NOT NULL
 2. STimeDate NOT NULL
 3. RoomNum NOT NULL
 4. LocID NOT NULL
 5. SeatNum NOT NULL

4) Functional Dependencies

NonMember:

R(Email, Name, PayMethod, PhoneNum, ChainID)

FDs:

Email -> Name, PayMethod, PhoneNum, ChainID

PhoneNum -> Email, Name, PayMethod, ChainID

RewMember:

R(Email, Name, PayMethod, PhoneNum, ChainID, RewPoints)

FDs:

Email -> Name, PayMethod, PhoneNum, ChainID, RewPoints

PhoneNum -> Email, Name, PayMethod, ChainID, RewPoints

FoodItem:

R(ItemSaleID, Price, Name, MemberEmail, TicketID)

FDs:

ItemSaleID -> Price, Name, MemberEmail, TicketID

Name -> Price

University of British Columbia, Vancouver

Department of Computer Science

Decomposition:

R(ItemSaleID, Price, Name, MemberEmail, TicketID)

Name->Price violates 3NF

R1(Name, Price)

R2(Name, ItemSaleID, MemberEmail, TicketID)

NonRecSeat:

R(SeatNum, RoomNum, LocID)

FDs: none

RecSeat:

R(SeatNum, RoomNum, LocID)

FDs: None

Auditorium:

R(RoomNum, LocID, NumRecSeats, NumNonRecSeats)

FDs: None

Theater:

R(LocID, PostalCode, Province, City, ChainID)

FDs:

LocID -> PostalCode, Province, City, ChainID

PostalCode -> Province, city

Decomposition:

R(LocID, PostalCode, Province, City, ChainID)

PostalCode -> Province, City violates 3NF

R1(PostalCode, Province, City)

R2(LocID, PostalCode, ChainID)

TheaterChain:

R(ChainID)

FDs: None

NormalTicket:

R(TicketID, NPrice, STimeDate, RoomNum, LocID, SeatNum, Email)

FDs:

TicketID -> NPrice, STimeDate, RoomNum, LocID, SeatNum, Email

GoldTicket:

R(TicketID, GPrice, STimeDate, RoomNum, LocID, SeatNum, Email, ItemSaleID)

FDs:

TicketID -> NPrice, STimeDate, RoomNum, LocID, SeatNum, Email, ItemSaleID

5) Normalization

11. NonMember

- a. NonMember(Email: Char(20), Name: Char(20), PayMethod: Char(20),
PhoneNum: Char(11), ChainID: Char(20))
- b. Specifications
 - i. Primary Key: Email
 - ii. Candidate Key(s): Email, PhoneNum
 - iii. Foreign Keys: ChainID
 - iv. Other Constraints:
 - 1. TheaterChain NOT NULL
 - 2. PhoneNum UNIQUE

12. RewMember

- a. RewMember(Email: Char(20), Name: Char(20), PayMethod: Char(20),
PhoneNum: Char(11), ChainID: Char(20),
RewPoints: Integer)
- b. Specifications
 - i. Primary Key: Email
 - ii. Candidate Key(s): Email, PhoneNum
 - iii. Foreign Keys: ChainID
 - iv. Other Constraints:
 - 1. TheaterChain NOT NULL
 - 2. PhoneNum UNIQUE

13. FoodPrice

- a. FoodPrice(Price: Integer, Name: Integer)
- b. Specifications
 - i. Primary Key: Name
 - ii. Candidate Key(s): Name
 - iii. Foreign Keys: none
 - iv. Other Constraints: none

14. FoodItem

- a. FoodItem(ItemSaleID: Integer, Name: Integer, MemberEmail: Char(20), TicketID: Integer)
- b. Specifications
 - i. Primary Key: ItemSaleID
 - ii. Candidate Key(s): ItemSaleID

iii. Foreign Keys: MemberEmail, TicketID

iv. Other Constraints:

1. TicketID UNIQUE

15. NonRecSeat

a. NonRecSeat(SeatNum: Integer, RoomNum: Integer, LocID: Integer)

b. Specifications

i. Primary Key: SeatNum, RoomNum, LocID

ii. Candidate Key(s): {SeatNum, RoomNum, LocID}

iii. Foreign Keys: RoomNum, LocID

iv. Other Constraints: none

16. RecSeat

a. RecSeat(SeatNum: Integer, RoomNum: Integer, LocID: Integer)

b. Specifications

i. Primary Key: SeatNum, RoomNum, LocID

ii. Candidate Key(s): {SeatNum, RoomNum, LocID}

iii. Foreign Keys: RoomNum, LocID

iv. Other Constraints: none

17. Auditorium

a. Auditorium(RoomNum: Integer, LocID: Integer, NumRecSeats: Integer,
NumNonRecSeats: Integer)

b. Specifications

i. Primary Key: RoomNum, LocID

ii. Candidate Key(s): none

iii. Foreign Keys: LocID

iv. Other Constraints:

1. NumRecSeats NOT NULL

2. NumNonRecSeats NOT NULL

18. TheaterProCity

a. Theater(PostalCode: Char(6), Province: Char(20), City: Char(20))

b. Specifications

i. Primary Key: PostalCode

ii. Candidate Key(s): PostalCode

iii. Foreign Keys: none

iv. Other Constraints: none

19. TheaterLocId

a. Theater(LocID: Integer, PostalCode: Char(6), ChainID: Integer)

b. Specifications

i. Primary Key: LocID

ii. Candidate Key(s): LocID

iii. Foreign Keys: ChainID

iv. Other Constraints:

1. ChainID NOT NULL

20. TheaterChain

- a. TheaterChain(ChainID: Integer)
 - b. Specifications
 - i. Primary Key: ChainID
 - ii. Candidate Key(s): ChainID
 - iii. Foreign Keys: none
 - iv. Other Constraints: none
21. NormalTicket
- a. NormalTicket(TicketID: Integer, NPrice: Integer, STimeDate: Char(20), RoomNum: Integer, LocID: Integer, SeatNum: Integer, Email: Char(20))
 - b. Specifications
 - i. Primary Key: TicketID
 - ii. Candidate Key(s): TicketID
 - iii. Foreign Keys: Email, STimeDate, RoomNum, LocID, SeatNum
 - iv. Other Constraints:
 - 1. Email NOT NULL
 - 2. STimeDate NOT NULL
 - 3. RoomNum NOT NULL
 - 4. LocID NOT NULL
 - 5. SeatNum NOT NULL
22. GoldTicket
- a. GoldTicket(TicketID: Integer, GPrice: Integer, STimeDate: Char(20), RoomNum: Integer, LocID: Integer, SeatNum: Integer, Email: Char(20), ItemSaleID: Integer)
 - b. Specifications
 - i. Primary Key: TicketID
 - ii. Candidate Key(s): TicketID
 - iii. Foreign Keys: STimeDate, RoomNum, LocID, SeatNum, ItemSaleID
 - iv. Other Constraints:
 - 1. Email NOT NULL
 - 2. STimeDate NOT NULL
 - 3. RoomNum NOT NULL
 - 4. LocID NOT NULL
 - 5. SeatNum NOT NULL

6) SQL tables

```
CREATE TABLE Customer_NonMember (  
    Email CHAR(20),  
    Name CHAR(20),  
    PaymentMethod CHAR(20),
```


University of British Columbia, Vancouver

Department of Computer Science

```
    PhoneNum CHAR(20),
    TheatreChainID CHAR(20) NOT NULL,
    UNIQUE (PhoneNum),
    PRIMARY KEY (Email),
    FOREIGN KEY (TheatreChainID) REFERENCES TheatreChain
);

CREATE TABLE Customer_Member (
    Email CHAR(20),
    Name CHAR(20),
    PaymentMethod CHAR(20),
    PhoneNum CHAR(20),
    TheatreChainID CHAR(20) NOT NULL,
    RewardsPoints INTEGER,
    UNIQUE (PhoneNum),
    PRIMARY KEY (Email),
    FOREIGN KEY (TheatreChainID) REFERENCES TheatreChain
);

CREATE TABLE FoodItem (
    ItemSaleID INTEGER,
    Name          INTEGER,
    MemberEmail CHAR(20),
    TicketID INTEGER,
    PRIMARY KEY (ItemSaleID),
    FOREIGN KEY (MemberEmail) REFERENCES Customer
    FOREIGN KEY (TicketID) REFERENCES GoldTicket
)

CREATE TABLE FoodPrice (
    Name CHAR(20),
    Price INTEGER,
    PRIMARY KEY (Name),
)

CREATE TABLE NonRecSeat (
    SeatNum INTEGER,
    RoomNum INTEGER,
    LocID INTEGER,
    PRIMARY KEY (SeatNum, RoomNum, LocID),
    FOREIGN KEY (RoomNum, LocID) REFERENCES (Auditorium, Theatre)
)
```

University of British Columbia, Vancouver

Department of Computer Science

```
CREATE TABLE RecSeat (  
    SeatNum INTEGER,  
    RoomNum INTEGER,  
    LocID INTEGER,  
    PRIMARY KEY (SeatNum, RoomNum, LocID),  
    FOREIGN KEY (RoomNum, LocID) REFERENCES (Auditorium, Theatre)  
)
```

```
CREATE TABLE MovieShowing (  
    MovieTitle CHAR(20) NOT NULL,  
    STimeDate CHAR(20) NOT NULL,  
    Runtime CHAR(20),  
    RoomNum INTEGER,  
    LocID INTEGER,  
    PRIMARY KEY (STimeDate, RoomNum, LocID),  
    FOREIGN KEY (RoomNum, LocID) REFERENCES (Auditorium, Theatre)
```

```
CREATE TABLE Auditorium (  
    RoomNum INTEGER,  
    LocID INTEGER,  
    NumRecSeats INTEGER NOT NULL,  
    NumNonRecSeats INTEGER NOT NULL,  
    PRIMARY KEY (RoomNum, LocID),  
    FOREIGN KEY (LocID) REFERENCES Theatre  
)
```

```
CREATE TABLE TheaterProCity (  
    PostalCode CHAR(6),  
    City CHAR(20),  
    Province CHAR(20),  
    PRIMARY KEY (PostalCode)  
)
```

```
CREATE TABLE Theater (  
    LocID INTEGER,  
    PostalCode CHAR(6),  
    ChainID INTEGER NOT NULL,  
    PRIMARY KEY (LocID)  
    FOREIGN KEY (ChainID) REFERENCES TheatreChain  
)
```

```
CREATE TABLE TheatreChain (  
    ChainID INTEGER,
```

University of British Columbia, Vancouver

Department of Computer Science

```
ChainID INTEGER,  
PRIMARY KEY (ChainID),  
)  
  
CREATE TABLE NormalTicket (  
    TicketID INTEGER,  
    NPrice INTEGER,  
    STimeDate CHAR(20) NOT NULL,  
    RoomNum INTEGER NOT NULL,  
    LocID INTEGER NOT NULL,  
    SeatNum INTEGER NOT NULL,  
    Email CHAR(20) NOT NULL,  
    PRIMARY KEY (TicketID)  
    FOREIGN KEY (Email, STimeDate, RoomNum, LocID, SeatNum)  
)  
  
CREATE TABLE GoldTicket (  
    TicketID INTEGER,  
    NPrice INTEGER,  
    STimeDate CHAR(20) NOT NULL,  
    RoomNum INTEGER NOT NULL,  
    LocID INTEGER NOT NULL,  
    SeatNum INTEGER NOT NULL,  
    Email CHAR(20) NOT NULL,  
    PRIMARY KEY (TicketID)  
    FOREIGN KEY (Email, STimeDate, RoomNum, LocID, SeatNum)  
)
```

7) TABLE EXAMPLE

Customer Non-Member						
Email	Name	TheatreC hainID	Phone	PaymentMe thod		
customer1@m ail.com	Customer1 Lastname	AMC	1111111	MasterCard		
customer2@m ail.com	Customer2 Lastname	IMAX	2222222	VISA		

University of British Columbia, Vancouver

Department of Computer Science

customer3@mail.com	Customer3 Lastname	AMC	3333333	MasterCard		
customer4@mail.com	Customer4 Lastname	IMAX	4444444	VISA		
customer5@mail.com	Customer5 Lastname	AMC	5555555	MasterCard		
Customer Rewards Member						
Email	Name	TheatreC hainID	Phone	PaymentMe thod	Rewards Points	
customer1@mail.com	Customer1 Lastname	AMC	1111111	MasterCard	3	
customer2@mail.com	Customer2 Lastname	IMAX	2222222	VISA	4	
customer3@mail.com	Customer3 Lastname	AMC	3333333	MasterCard	3	
customer4@mail.com	Customer4 Lastname	IMAX	4444444	VISA	2	
customer5@mail.com	Customer5 Lastname	AMC	5555555	MasterCard	3	
Food Item						
ItemSaleID	MemberEmail					
1	custmer1@mail.com					
2	custmer2@mail.com					
3	custmer3@mail.com					
4	custmer4@mail.com					
5	custmer5@mail.com					
FoodPrice						

University of British Columbia, Vancouver

Department of Computer Science

Name	Price (\$)					
Burger	1					
Fries	2					
Chips	3					
Potatoes	4					
Popcorn	5					
Non-Recliner Seat						
SeatNum	RoomNum	LocID				
1	1	1				
2	2	2				
3	3	3				
4	4	4				
5	5	5				
Recliner Seat						
SeatNum	RoomNum	LocID				
6	1	1				
7	2	2				
8	3	3				
9	4	4				
10	5	5				
MovieShowing						
Title	Date and Time	Runtime	RoomNum	LocID		
Movie 1	April 1, 2021 2:00 PM	120 min	1	1		
Movie 2	April 1, 2021 3:00 PM	120 min	3	2		
Movie 3	April 1, 2021 4:00 PM	120 min	4	2		
Movie 4	April 1, 2021 5:00 PM	120 min	1	2		

University of British Columbia, Vancouver

Department of Computer Science

Movie 5	April 1, 2021 6:00 PM	120 min	2	1		
Auditorium						
RoomNum	LocID	NumRecS eats	NumNormal Seat			
1	1	30	120			
2	1	30	300			
3	1	30	300			
1	3	30	300			
1	2	30	300			
Theatre						
LocID	Postal Code	ChainID				
1	YVR AR1	AMC				
2	YVR AR2	AMC				
3	YVR AR3	IMAX				
4	YVR AR4	IMAX				
5	YVR AR5	AMC				
Theatre Chain						
ChainID						
AMC1						
AMC2						
AMC3						
IMAX1						
IMAX2						
Normal Ticket						
TicketID	Price	Showing Time	RoomNum	LocID	SeatNum	Email
1	\$4	April 1, 2021 2:00 PM	1	1	1	customer1@mail.c om

University of British Columbia, Vancouver

Department of Computer Science

2	\$4	April 1, 2021 3:00 PM	2	2	2	customer2@mail.c om
3	\$4	April 1, 2021 4:00 PM	3	3	3	customer3@mail.c om
4	\$4	April 1, 2021 5:00 PM	4	4	4	customer4@mail.c om
5	\$4	April 1, 2021 6:00 PM	5	5	5	customer5@mail.c om
Gold Ticket						
TicketID	Price	Showing Time	RoomNum	LocID	SeatNum	Email
1	\$7	April 1, 2021 2:00 PM	1	1	1	customer1@mail.c om
2	\$7	April 1, 2021 3:00 PM	2	2	2	customer2@mail.c om
3	\$7	April 1, 2021 4:00 PM	3	3	3	customer3@mail.c om
4	\$7	April 1, 2021 5:00 PM	4	4	4	customer4@mail.c om
5	\$7	April 1, 2021 6:00 PM	5	5	5	customer5@mail.c om

8) List of queries

- “Insertion: Add movie showings to the table list of movie showings.”
- “Insertion: Add customer to list of rewards members.”
- “Insertion: Add food item purchase.”
- “Insertion: Add ticket purchase.”
- “Update: The number of rewards a rewards member has.”
- “Delete: Movie showing that has already been played is deleted.”
- “Select: Movie showing to attend.”

- “Select: Auditorium seat (recliner or non-recliner).”
- “Projection: Ticket and movie showing. Movie date, title and runtime in the projection condition.”
- “Join: Movie ticket and movie showtime.”
- “Aggregation Group by: Number of movie seats in theatre auditorium, recliner and non-recliner, and availability for gold and normal tickets.”
- “Aggregation with Having: HAVING clause selects from group. Select movie showing from the theatre location.”
- “Nested Aggregation with Group By: Find popularity of movie showing and of seats within a theatre auditorium by finding the average number of customers that purchase that seat or a ticket to that movie showing.”
- “Division: Find all customers who attended a movie showing.”