**Database Specification**

Database Management System: MySQL

Parent tables:

1. Customer:
2. Customer\_id(PK,VARCHAR(16))
3. Fname(VARCHAR(25))
4. Lname(VARCHAR(25))
5. DOB(DATE)
6. Phone\_number(CHAR(10))
7. Email(VARCHAR(50))
8. Username(VARCHAR(25))
9. Password(VARCHAR(50))
10. Theater\_Location:
11. Theater\_id(PK,VARCHAR(16))
12. Name(VARCHAR(25))
13. Address(VARCHAR(50))
14. State(CHAR(2))
15. Zip\_Code(CHAR(5))

Child tables:

1. Payment:
2. Transaction\_number(PK, VARCHAR(16))
3. Customer\_id(FK,VARCHAR(16))
4. Bank(VARCHAR(25))
5. Card\_number(VARCHAR(19))
6. Ticket\_id(FK,VARCHAR(16))
7. Tickets
8. Ticket\_id(PK,VARCHAR(16))
9. Movie\_name(VARCHAR(50))
10. Room\_id(FK,VARCHAR(16))
11. Movie\_rating(VARCHAR(5))
12. Date(DATE)
13. Showtime(TIME)
14. Theater\_id(FK, VARCHAR(16))
15. Showtimes
16. Showtime\_id(PK,VARCHAR(16))
17. Theater\_id(FK,VARCHAR(16))
18. Showtime(TIME)
19. Room\_id(FK,VARCHAR(16))
20. Date(DATE)
21. Seats\_available
22. Room\_id(PK,VARCHAR(16))
23. Room\_number(VARCHAR(3))
24. Seat\_number(VARCHAR(3))
25. Customer\_id(FK,VARCHAR(16))
26. Ticket\_id(FK,VARCHAR(16))
27. Theater\_id(FK,VARCHAR(16))
28. Seats\_remaining(VARCHAR(3))

Specification:

    The customer table will be a parent table used to store everything related to the customer. This is used for login purposes, texting or emailing tickets, and for tracking the tickets they purchase. This is done so that selected theaters can properly adjust how many tickets are left for a movie at a certain showtime.

    The theater table is another parent table that is used to store information about every theater location. Each theater has a unique I.D and location information.

    The payment table is a child table that will be used to track each customer's purchases. For this reason the “customer\_id” is listed as a foreign key. This allows for a join when performing queries on the database to make the customers purchases identifiable. It also lists the “ticket\_id” as a foreign key because each ticket sold has its own identifier. This allows a join with the ticket table that allows us to find the movie, room\_id, showtime, and theater\_id.

    The tickets table is a child table that stores all the information about the movie being seen. It has the name of the movie, the room number, the showtime, the date, and the theater\_id. The “theater\_id” is listed as a foreign key, which allows for a join, and gives access to the information about where the theater is located and the name of the theater.

    The showtimes table is a child table that stores all the showtimes associated with a given date at a particular theater. To allow this to happen, the theater\_id is listed as a foreign key. This allows for a join and the database can store all the appropriate information about the movie times at that theater. It also stores the room\_id as a foreign key. This allows a join on the seats remaining table and uniquely identifies a room in a theater. This allows the database to track how many seats are available that when the room is full, the showtime is removed from the available times.

    The seats available table is also a child table that stores information about the seats in a room at a particular theater. By listing the theater\_id, customer\_id, and ticket\_id as foreign keys in this table, it allows the database to reserve seating at a given theater location for a customer with the given ticket. This is done by allowing the tables to be joined. It also tracks this for each showtime because this table can be joined with the showtimes table. This helps keep track of the seats remaining.