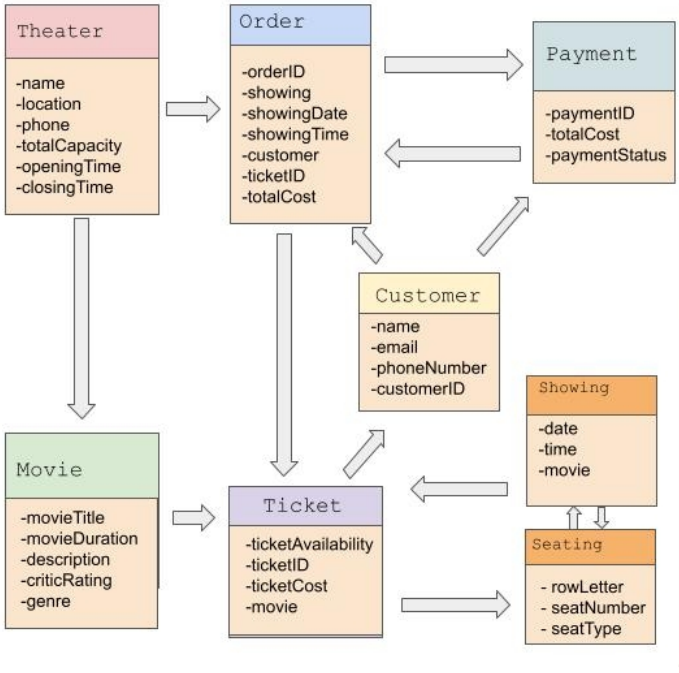
Architecture Design and Data Management

Software Architecture Diagram



Data Dictionary:

Theater:
location: address of theater
totalCapacity: max # of people theater can fit
opening/closingTime: hours of the theater

Movie:
movieTitle: name of the movie
movieDuration: length of the movie
description: short movie description
criticRating: movie rating

Ticket:
ticketID: ticket identifier for new orders
ticketCost: price of the ticket (before tax)
movie: movie the ticket was created for

Order:
orderID: unique ID generated when order is made
showing: what movie the order is for
showingDate/Time: day and time of showing
ticketID: same ticket identifier
totalCost: price listed before payment is made

Showing:
date: available dates of a showing
time: available times for a showing

Seating:
rowLetter/seatNumber: location of seat
seatType: standard seat or disability seat?

Payment:
paymentID: key associated with a particular payment transaction
totalCost: total cost of order after tax
paymentStatus: bool if payment was successful

Data Management Strategy:

Theaters

id	Theater	Screens	State	City
00001	Theater #1	18	California	San Diego

Movies

id	Title	Description	Durtaion	Rating
1	"Iron Man"		126	4.9/5.0
2	"Iron Man 2"		124	4.0/5.0
3	"Iron Man 3"		131	4.2/5.0

Showtimes

id	movie_id	theater_id	Price	start_time	end_time
00004	1	00001	$10.99	7:00pm	9:30pm
00005	2	00001	$10.99	2:00pm	4:30pm
00006	3	00001	$10.99	5:00pm	7:30pm

Customers

id	first_name	last_name	email	phone_number
00007	Gabriel	Noda	gnoda@gmai.com	619-726-6283
00008	Cade	Harbin	charbin@gmail.com	619-348-2233
00009	Francisco	Ortiz	fortiz@gmail.com	619-773-9090

Tickets

id	showtime_id	customer_id
00010	00004	00007
00011	00005	00008
00012	00006	00009

Payments

id	ticket_id	payment_method	ticket_quantity	total_price
00013	00010	Paypal	2	$21.98
00014	00011	Navy Federal	3	$32.97
00015	00012	Discover	2	$21.98

Our Ticketing system uses one database using the SQL Data Management strategy. Our database stores data of the theater ticketing system, such as customer information, ticket purchases, and showtime information. Each table represents a single entity and attributes, which for some create a relationship between the entities. We used SQL because the data is related, is organized and structured which helps us store data effectively, and because SQL is often better with run-time complexity, and we wouldn't want customers to struggle with a slow interface and system. Using one database seemed beneficial since the information we are looking to store is related, and is convenient to have centralized in order for simplicity of human oversight and would have a lower maintenance cost to access. Some tradeoffs to using a one database system compared to a multiple database system would be time complexity, maintenance cost, and overall convenience. If there were a vast amount of theaters and seats, a multiple database system would be more organized. If we decided to use NoSQL, we would be able to adapt our system to practices like Agile Scrum, and we would have easier scalability of the system, but using NoSQL would have trouble handling financial transactions, which is important in a theater.