concessions

	concession_id	SERIAL
FK	customer_id	INTEGER
	concession_amount	VARCHAR
	concession_date	VARCHAR

customers

PK	customer_id	SERIAL
	first_name	VARCHAR
	last_name	VARCHAR
	email	VARCHAR
FK	movie_id	INTEGER
FK	concession_id	INTEGER
FK	game_id	INTEGER
FK	membership_id	INTEGER

arcade

PK	game_id	SERIAL
	game_name	VARCHAR
	times_played	INTEGER
	tickets_given_out	INTEGER
FK	customer_id	INTEGER

membership

PK	membership_id	SERIAL
FK	customer_id	INTEGER
	end_date	DATE
	member_since	DATE

tickets

	payment_id	SERIAL
FK	customer_id	INTEGER
FK	movie_id	INTEGER
	ticket_amount	VARCHAR
	payment_date	VARCHAR

movie

PK	movie_id	SERIAL
	title	VARCHAR
	description	VARCHAR
	length	VARCHAR
	rating	VARCHAR

1
From **movie** to **tickets**, I used a "one to zero-or-many" relationship for **movie_id** because one movie could have sold zero tickets, but it also could have sold many tickets.

2
From **customers** to **tickets**, I used a "one to zero-or-many" relationship for **customer_id** because there could be a customer who only buys concessions and buys zero tickets. But, a customer could also buy several tickets like for a group of friends/family.

3
From **customers** to **concessions**, I used a "one to zero-or-many" relationship for **customer_id** because one customer could only buy a movie ticket and buy zero concessions. But, a customer could also buy several conessions if they're pretty hungry.

4
From **customers** to **membership**, I used a "one to zero-or-many" relationship for **customer_id** because one customer might not have a membership, but that customer could also be paying for multiple memberships like a family plan.

5
From **customers** to **movie**, I used a "one to zero-or-many" relationship for **movie_id** because one customer could have seen zero movies (maybe just bought food or played in the arcade), but that same customer also could have seen many movies.

6
From **customers** to **membership**, I used a "one-and-only-one to one-and-only-one" relationship for **membership_id** because each membership is tied to exactly one person, zero people cannot be tied to a membership nor can many. It's a one-to-one relationship.

7
From **customers** to **arcade**, I used a "one to zero-or-many" relationship for **customer_id** because one person can play either zero games or many games.

8
From **arcade** to **customers**, I used a "one to zero-or-many" relationship for **game_id** because one customer could have played zero games or many games.