

Customers		
PK	customer_id	serial
	last_name	varchar(40)
	payment_method	varchar(40)
	obnoxious_level	varchar(100)

Movies		
PK	movie_title	varchar(150)
	movie_length	integer
	rotten_tomatoes_rating	numeric (13,10)
	number_of_nude_scenes	integer

Concessions		
PK	food_type	varchar(50)
	food_price	numeric (5, 2)
	calories	numeric (6,2)
	vegan_friendly	varchar (3)

Tickets		
PK	amount	numeric (5,2)
	old_fart_discount_amount	numeric (5,2)
	number_tickets	integer
FK	customer_id	integer
FK	movie_title	varchar(150)

One customer could see many movies and one movie can be attended by many customers? Should we focus just on the Primary Keys when trying to figure out relationships

I think this should be many to many on both ends, but Lucid wouldn't obey me

My reasoning is that one customer could eat many different food types and the a given food type could be eaten by many different customers

I think this should be one to many since a customer might buy multiple tickets, but each ticket would only belong to one customer

Lucid was mean to me here too. I think this would be many to many because one movie could be seen for different amounts/prices because there could be adult, child, old fart. And all movies would have the same amount choices