Mock Theater ERD

customer table with customer info that uses a customer_id as primary key

customer		
PK	customer_id	SERIAL
	first_name	VARCHAR(100)
	last_name	VARCHAR(100)
	address	VARCHAR(150)
	billing_info	VARCHAR(150)

concession		
PK	sale_id	SERIAL
FK	customer_id	INTEGER
FK	product_number	INTEGER
	sale_amount	NUMERIC(5,2)
	quantity_sold	INTEGER

concession table for keeping track of consession sales that uses a sale_id as the primary key. It is linked to customer and inventory by FKs of customer_id, to track who bought what and product_number to update inventory.

inventory		
PK	product_number	INTEGER
	quantity	INTEGER

inventory table for keeping track of how many of each concession product is in stock. When implemented, the qantity should be decreased everytime something is sold

ticket		
PK	ticket_number	SERIAL
FK	theater_number	INTEGER
FK	movie_id	INTEGER
	ticket_cost	NUMERIC(4,2)
FK	customer_id	INTEGER
	sold_for_show	INTEGER

theater		
PK	theater_number	INTEGER
	capacity	INTEGER

theater table for tracking how many tickets for each show can be sold that uses a theater_number as the primary key. When implemented, the "sold_for_show" row from ticket would not be allowed to exceed the capcaity in this table

ticket table that uses ticket_number as a primary key. It records the ticket that is sold and uses FKs of theater_number, movie_id and customer_id. The FKs of movie_id and customer_id are to link those tables to get more info from each. The theater_number would work together with "sold_for_show" in this table and "capcaity" in the theater table to prevent overselling the show

movie		
PK	movie_id	SERIAL
	movie_name	VARCHAR(100)
	runtime	VARCHAR(10)

table containing infomation about the movie that uses a movie_id as a primary key