

## Problem statement:

We aim to develop a web-based management system for a car rental company. It has two main functionalities:

1. Facilitate the rental company to manage their car fleet. For example: the rental company can add a new car to their fleet, update a car's rental price and other information, or delete an existing car from the fleet.
2. Provide a powerful tool for the rental company to monitor their business activities in all aspects. For example: the system can display all the cars owned by the rental company, fleets at different locations, amount of transactions per month, number of customers etc. By closely monitoring their business data, the rental company can adjust business strategies based on performance as well as gain a better understanding of their customers in order to promote the quality of service.

Specially, the application will support:

1. CRUD car entity (add new car to fleet, update car rental price, delete car from fleet)
2. Query/filter cars that company owns by car type, year of service, mileage etc.
3. Query/filter car rental transactions and filter by date range, car type, location
4. Query/filter customer information by car type, rental history
5. Yearly statistics (total car rent, total number of transaction, monthly trend)

## Business rules:

1. The rental company owns many different rental locations.
2. Each rental location can manage multiple different cars, but one car can only be located at one location at a time.
3. Customers rent a car, which they pickup from one rental branch and return it at another rental branch (not necessarily the same).
4. Every car can be rented by many different customers at different times.
5. Each customer can rent many different cars at different times.
6. Each customer can conduct multiple transactions with the rental company, but each transaction can only belong to one customer.
7. Each car has its own car make and car model info.

## Nouns:

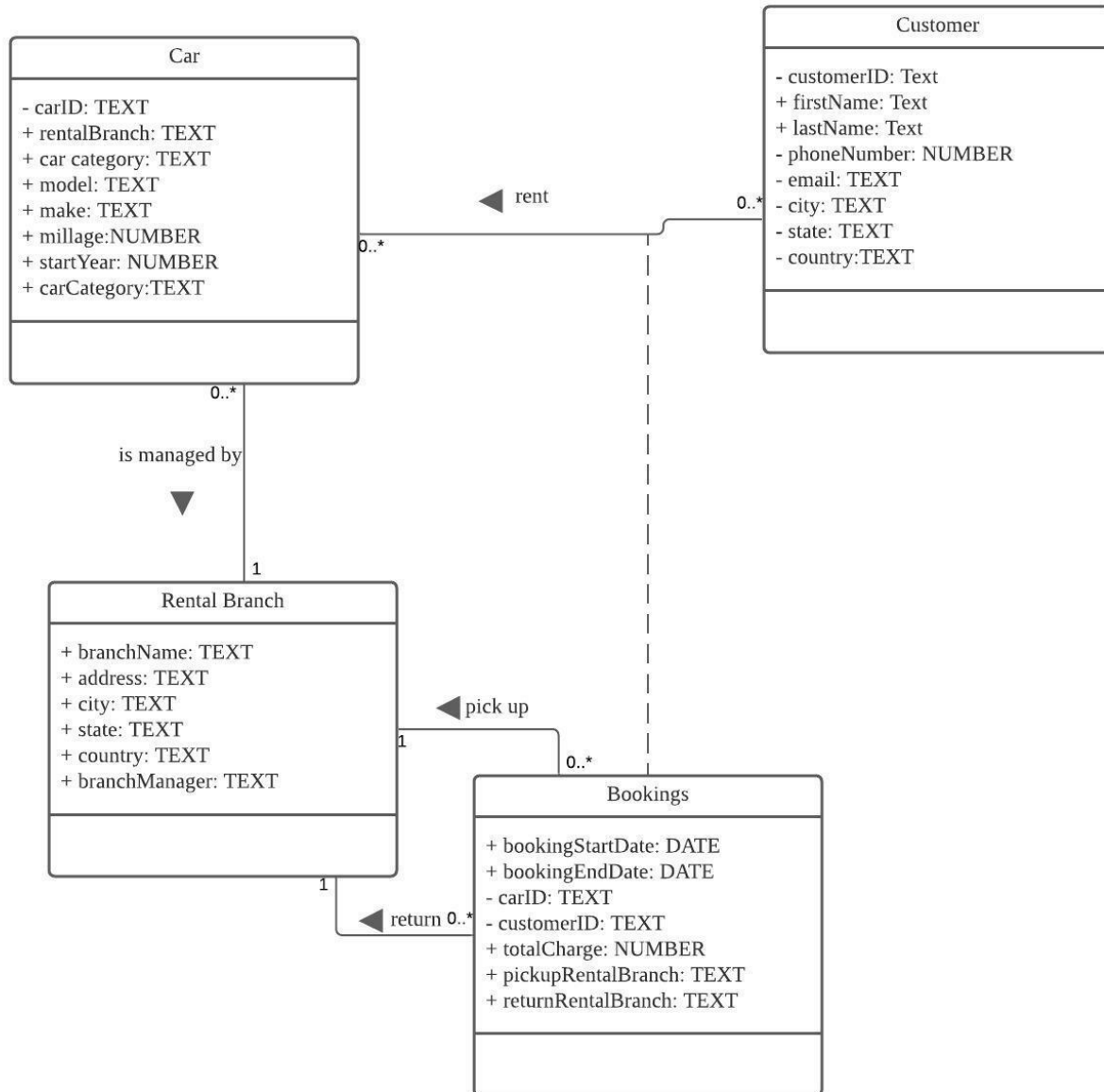
***Car, Rental Branch, Customers, Bookings, Car Make, Car Model***

## Verbs:

***Owns (rental branch), manage (cars), rented (by customers), rent(car), conduct (Bookings), has (car make/car model)***

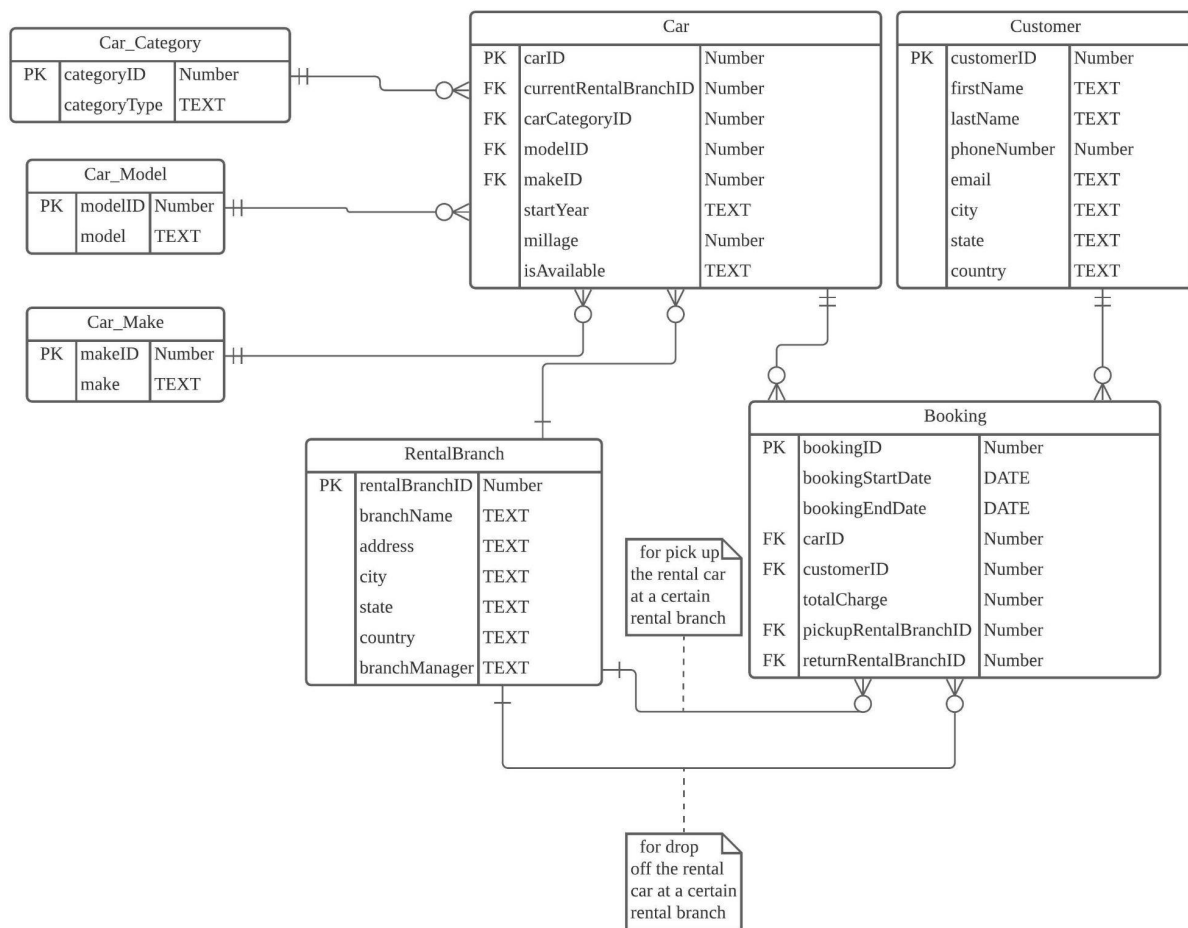
## UML Diagram:

[https://lucid.app/lucidchart/722ed355-65d1-47df-9cc8-3ba5faa57deb/edit?beaconFlowId=341BAFE5E164E51C&invitationId=inv\\_6ba983b6-83d2-4e74-866d-5fbff519bbd8&page=0\\_0#](https://lucid.app/lucidchart/722ed355-65d1-47df-9cc8-3ba5faa57deb/edit?beaconFlowId=341BAFE5E164E51C&invitationId=inv_6ba983b6-83d2-4e74-866d-5fbff519bbd8&page=0_0#)



## ERD:

[https://lucid.app/lucidchart/4cc47107-7d20-475b-9944-03504518d9ac/edit?beaconFlowId=E9C9B42E284D081A&invitationId=inv\\_9cbd9f0a-f83d-4c95-931b-c6932ece15a2&page=0\\_0#](https://lucid.app/lucidchart/4cc47107-7d20-475b-9944-03504518d9ac/edit?beaconFlowId=E9C9B42E284D081A&invitationId=inv_9cbd9f0a-f83d-4c95-931b-c6932ece15a2&page=0_0#)



## BCNF:

Our database schema:

Car(carID, currentRentalBranchID, carCategoryID, modelID, makeID, startYear, mileage, isAvailable)

Customer(customerID, firstName, lastName, phoneNumber, email, city, state, country)

Booking(bookingID, bookingStartDate, bookingEndDate, carID, customerID, totalCharge, pickupRentalBranchID, returnRentalBranchID)

RentalBranch(rentalBranchID, branchName, address, city, state, country, branchManager)

Car Category(categoryID, categoryType)

Car Model(modelID, model)

Car Make(makeID, make)

Proof that our schemas are in BCNF:

1. They are in 1NF because all the attributes in all schemas are single-valued.
2. To prove the schema is in 2NF and BCNF, list out all the functional dependencies in each schema:

Car:

carID -> currentRentalBranchID

carID -> carCategoryID

carID -> modelID

carID -> makeID

carID -> startYear

carID -> mileage

carID -> isavailable

There's only one primary key in this table. All the attributes in the Car schema are only and fully dependent on the primary key alone. Therefore this schema is in 2NF and BCNF.

Customer:

customerID -> firstName

customerID -> lastName

customerID -> phoneNumber

customerID -> email

customerID -> city

customerID -> state

customerID -> country

There's only one primary key in this table. All the attributes in the Customer schema are only and fully dependent on the primary key alone. Therefore this schema is in 2NF and BCNF.

Booking:

bookingID -> bookingStartDate

bookingID -> bookingEndDate

bookingID -> carID

bookingID -> customerID

bookingID -> totalCharge

bookingID -> pickupRentalBranchID

bookingID -> returnRentalBranchID

There's only one primary key in this table. All the attributes in the Booking schema are only and fully dependent on the primary key alone. Therefore this schema is in 2NF and BCNF.

RentalBranch:

rentalBranchID -> branchName

rentalBranchID -> address

rentalBranchID -> city

rentalBranchID -> state

rentalBranchID -> country

rentalBranchID -> branchManager

There's only one primary key in this table. All the attributes in the RentalBranch schema are only and fully dependent on the primary key alone. Therefore this schema is in 2NF and BCNF.

Car Category:  
categoryID -> categoryType

Car Model:  
modelID -> model

Car Make:  
makeID -> make

There's only one primary key in these tables. All the attributes in these three schemas are only and fully dependent on their respective primary key alone. Therefore, these schemas are in 2NF and BCNF.