

# Data Management and Database Design

## **CAR RENTAL DATABASE MANAGEMENT SYSTEM**

Contents

Team Members.....3

Project Topic.....3

Problem Statement.....3

Objectives.....3

Proposed Solution for identified problem.....4

Entity Relationship Diagram.....5

Database Dictionary.....6

## Team Members

Sr. No.	Name	Email Id
1	Bhawna Singh	singh.bhaw@northeastern.edu
2	Yashwanth Kolla	kolla.y@northeastern.edu
3	Shereen John	john.sh@northeastern.edu
4	Uday Kumar Reddy Somireddy	somireddy.u@northeastern.edu
5	Sanket Khadke	khadke.s@northeastern.edu
6	Madhava Potturu	potturu.m@northeastern.edu

## Project Topic

### Car Rental Database Management System

## Problem Statement

- For many people, buying a car and bearing its insurance cost would be expensive
- For a traditional car rental service, a customer must physically visit the rental store which is inefficient
- In our car rental system, a user can book a car ride from any place conveniently, and after the payment is made, s/he will be notified of the nearest location from where s/he can pick up the car
- Due to high availability of these locations, the user can get to these locations in a short duration of time
- User can then use the car for the specified session and then return it to one of the closest locations

## Objectives

To design a database for the car rental platform which should support the following functionalities:

- Users must be able to register into the system with details such as username, email, address, etc., and then login using those credentials
- Users must be able to view the current availability of cars for rent in his/her nearby location
- Users must be able to book a car followed by the payment of the session
- Ride details must be stored, and users must be able to view the booking details
- Maintain details of car location such as parking spot, area name, zip code, etc.

## Proposed Solution for the identified problems

### 1. Difficulty in tracking daily transactions and increased chance of error and complexity in maintaining them

To overcome the above problem, a database management system is created for the car rental system, where instead of maintaining the data manually, the data will be logged in its respective entity tables. This will facilitate efficient managing of the structured data. Objects such as indexes, triggers, views, etc. would be used which would fine tune the system as well as maintain the ETL process of the database.

### 2. Same data is being entered daily, which is a redundant process (car, location data)

Redundancy is being eliminated by:

- i. Passenger data need not be entered manually every time the passenger books a car for rent. All the required customer data will be stored in the "passenger" table.
- ii. All the required details about cars available in the rental system will be stored in the tables "car" and "car category".
- iii. Database will ensure that at a time, no two reservations on the same car are being made.
- iv. Tables will be normalized to ensure data consistency and uniqueness.

### 3. No database available for the passenger, car, and the location of car spots

Three different tables for passenger, car and location have been created. Each table, **passenger**, **car**, and **location** have their unique ids, as **passenger\_id**, **license\_no** and **location\_id**, respectively. These unique ids will help determine each of the passengers, car, and location uniquely.

### 4. There is no system of booking a car for a certain time online

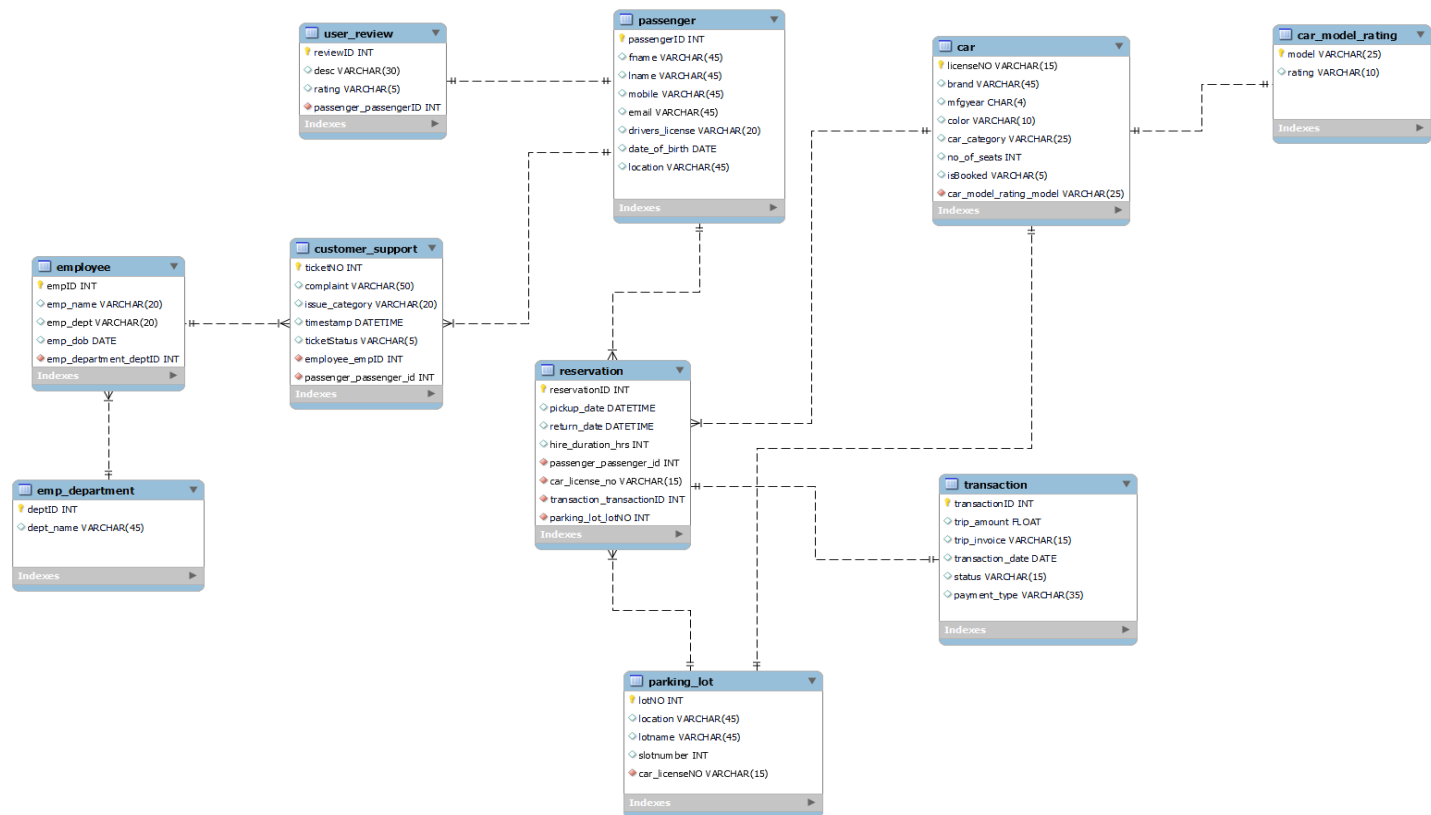
"Reservation" entity is created in the database to book a car online, data in this table has **reservation\_id** attribute as primary key to maintain uniqueness. The reservation table has the attributes **passenger\_id**, **car\_id** and **parking\_lotno** as foreign keys that specify what car a passenger has booked and in which parking lot would he/she find the required car.

## 5. Audit of transactions done by customer bookings is difficult

With the creation of the database, auditing of past transactions is easy to track.

The table 'transaction' keeps the records of all the transactions done. The attribute **transaction\_id** identifies a specific transaction, the foreign key **reservation\_id** stores the ID of the reservation which was done by a particular passenger. It also stores information in attributes like **trip\_amount**, **trip\_invoice**, and **transaction\_date** which stores the total trip amount, the invoice number generated, and the date of transaction, respectively.

## Entity – Relationship Diagram



## Database Dictionaries

### Database details for Car Rental System

#### *Passenger*

Field name	Description	Type	Length	Comments
passengerID	Unique ID assigned to each passenger	INT		Primary Key
fname	Passenger's first name	VARCHAR	45	Not Null
lname	Passenger's last name	VARCHAR	45	Not Null
mobile	Mobile number of the passenger	VARCHAR	45	Not Null
email	Passenger's email id	VARCHAR	45	Not Null
drivers_license	Passenger's driver's license number	VARCHAR	20	Not Null
date_of_birth	Passenger's date of birth	DATE		Not Null
location	Passenger's location	VARCHAR	45	Not Null

#### *Reservation*

Field name	Description	Type	Length	Comments
reservationID	Unique ID assigned to each reservation made	INT		Primary Key
pickup_date	Time when the ride starts	DATETIME		Not Null
return_date	Time when the ride ends	DATETIME		Not Null
hire_duration_hrs	Time duration of the ride	INT		Not Null
car_license_no	Foreign key			Not Null
passenger_id	Foreign key			Not Null
lotNO	Foreign key			Not Null
transactionID	Foreign key			Not Null

### ***Car***

Field name	Description	Type	Length	Comments
licenseNO	License number	VARCHAR	15	Primary Key
brand	Make of car	VARCHAR	45	Not Null
mfgyear	Manufactured year	CHAR	4	Not Null
color	Car color	VARCHAR	10	Not Null
car_category	Car category	VARCHAR	25	Not Null
no_of_seats	Total no. of seats available	INT		Not Null
isBooked	Is the car booked	VARCHAR	5	Not Null
model	Foreign Key			Not Null

### ***Transaction***

Field name	Description	Type	Length	Comments
transactionID	Transaction ID	INT		Primary Key
trip_amount	Trip amount	FLOAT		Not Null
trip_invoice	Trip invoice	VARCHAR	15	Not Null
transaction_date	Transaction date	DATE		Not Null
status	Status of transaction	VARCHAR	15	Not Null
payment_type	Payment type	VARCHAR	35	Not Null

### ***Customer\_support***

Field name	Description	Type	Length	Comments
ticketNO	Ticket ID identifying ticket uniquely	INT		Primary Key
complaint	Description of the complaint	VARCHAR	50	Not Null
issue_category	The category to which the issue belongs	VARCHAR	20	Not Null
timestamp	The time at which the ticket was created	DATETIME		Not Null
ticketStatus	Whether the ticket is open or closed	VARCHAR	5	Not Null
emplID	Foreign key			Not Null
passenger_id	Foreign key			Not Null



### ***Parking lot***

Field name	Description	Type	Length	Comments
lotNO	Parking lot number	INT		Primary Key
location	Parking location	VARCHAR	45	Not Null
lotname	Parking lot name	VARCHAR	45	Not Null
slotnumber	Parking slot number	INT		Not Null
licenseNO	Foreign Key		15	Not Null

### ***User Review***

Field name	Description	Type	Length	Comments
reviewID	Unique ID assigned to each review	INT		Primary Key
desc	Passenger's review	VARCHAR	30	Not Null
rating	Passenger's rating	VARCHAR	5	Not Null
passengerID	Foreign key			Not Null

### ***Employee***

Field name	Description	Type	Length	Comments
empID	The ID of employee.	INT		Primary Key
emp_name	The name of the employee	VARCHAR	20	Not Null
emp_dept	The department to which the employee belongs	VARCHAR	20	Not Null
emp_dob	Employee's date of birth	DATE		Not Null
deptID	Foreign key			Not Null

### ***Emp\_department***

Field name	Description	Type	Length	Comments
deptID	The department ID of employee	INT		Primary Key
dept_name	The department name of the employee	VARCHAR	45	Not Null

***Car\_model\_rating***

Field name	Description	Type	Length	Comments
model	Car model	VARCHAR	25	Primary Key
rating	Rating for the car model	VARCHAR	10	Not Null