Car Rental DBMS

TABLE OF CONTENTS

Introduction **3**

Purpose **3**

Entities **4**

Relationships **7**

Constraints **8**

Example Queries **11**

Example Updates **11**

Users **11**

Conclusion **12**

Introduction

The Car Rental DBMS will be designed to organize customer, vehicle, reservation, rental, and payment data efficiently and effectively. It also tracks vehicle maintenance to ensure customer safety and satisfaction when making a rental. Using a DBMS for this scenario is vital to keep track of operations among thousands of customers and avoid redundancy.

Purpose

This report aims to describe the Car Rental DBMS, highlighting the application's functionality, the data/information it will store, relationships between entities, constraints, example queries and updates, and potential users. This report will serve as a guideline for the later design and development stages of the database system. It will outline the expectations that the Car Rental DBMS aims to meet to ensure a good experience for the customers that make use of this service. This DBMS is being created to ensure that reservations and vehicle rentals are made appropriately under controlled redundancy compared to traditional file processing systems. End users of this DBMS will be able to access their role specific view and perform queries/transactions to their needs, outlined further in this report.

Entities

1. Customer

   This entity represents the user of the rental application who has made a reservation/rental for the vehicle. This table will include related information to the customer such as customer ID, name, address, phone number, drivers license, and their email. The customer's name will be stored as a string of alphabetic characters. Customer address, phone number, drivers license and email will be stored as an alphanumeric string. The customer ID will be stored as an integer.

   CUSTOMER

CustomerID	Name	PhoneNumber	Email	Address	DriversLicense

2. Review

   This entity represents the reviews that customers leave behind after the car rental service. This table will include related information to the review such as review ID, review date, rating, and feedback. Review ID and rating will be stored as an integer. Feedback will be stored as a string of alphabetic characters and review date will be of date type.

   REVIEW

ReviewID	ReviewDate	Rating	Feedback

3. Vehicle

   This entity represents each vehicle that the rental company has. This table will include information such as vehicle ID, car model, license plate number, type (economy, luxury), daily rate, colour, and fuel capacity. Vehicle ID will be stored as an integer. Car model, type, status, and colour will be a string of alphabetical characters. The plate number will be an alphanumeric string. Daily rate and fuel capacity will be a float.

VEHICLE

VehicleID	Model	Plate	VType	DailyRate	Fuel Capacity	VStatus	Colour

4. Reservation

This entity represents each reservation made by a customer for a specific vehicle. This table will include information such as reservation ID, reservation start date, reservation end date, and reservation status (cancelled, confirmed, etc.). Reservation ID will be stored as an integer. Planned pickup and return date will be of date type. Reservation status will be a string of alphabetical characters.

RESERVATION

ReservationID	PPickupDate	PReturnDate	RStatus

5. Rental

This entity represents the actual rental details once the customer has picked up and returned the vehicle. This table will include information such as rental ID, picked up date, returned date, rental cost, fuel charge and level. Rental ID and fuel level will be stored as an integer. Pickup and return date will be of date type. Rental cost, and fuel charge will be a float.

RENTAL

RentalID	PickupDate	ReturnDate	RentalCost	FuelCharge	FuelLevel

6. Payment

This entity represents the financial transaction made for the rental. This table will include information such as payment ID, payment date, amount, and payment method (Cash, Debit, Credit). The payment method would be stored as a string of alphabetical characters, while the payment ID will be an integer. The payment date will be of date type and the total amount will be a float.

PAYMENT

PaymentID	PaymentDate	PaymentMethod	TotalAmount

7. Maintenance - SuperClass

This entity represents all the service and repairs made to vehicles. The table will include information such as maintenance ID, service date, and status. This table would act as a measure of safety before vehicles are put into the system for rentals. ID's would be stored as an integer, whereas date will be of date type. The status field would indicate if the vehicle is currently undergoing maintenance (in service or completed), stored as a string of alphabetical characters. Note that this is different from the status field of the vehicle table.

MAINTENANCE

MaintenanceID	ServiceDate	MStatus

8. Cleaning - SubClass

This is a specialized entity that represents the specific maintenance of cleaning. The table will include information such as cleaning area(interior, exterior) and detail level(basic, deepcleaning). Both of these attributes will be stored as a string of alphabetical characters. Note that this subclass entity will also inherit the attributes of its parent entity, maintenance.

CLEANING

CleaningArea	DetailLevel

9. Repair - SubClass

   This is a specialized entity that represents the specific maintenance of repairs. The table will include information such as repair type and parts repaired. Both of these attributes will be stored as a string of alphabetical characters. Note that this subclass entity will also inherit the attributes of its parent entity, maintenance.

REPAIR

RepairType	Part

Relationships

a) Each CUSTOMER record can be related to multiple RESERVATION records and therefore multiple RENTAL records

b) Each CUSTOMER record can be related to multiple REVIEW records

c) Each REVIEW record is related to one CUSTOMER record, and one VEHICLE record

d) Each RESERVATION record is related to one CUSTOMER record, one VEHICLE record, and one RENTAL record

e) Each VEHICLE record can be related to multiple RESERVATION records and thus, multiple RENTAL records as well

f) Each VEHICLE record can be related to multiple MAINTENANCE records

g) Each RENTAL record is related to one RESERVATION record and one PAYMENT record

h) Each PAYMENT record is related to one RENTAL record

i) Each MAINTENANCE record is related to one VEHICLE record

j) Each MAINTENANCE record is related to either one CLEANING record or one REPAIR record

Constraints

Key Constraints:

a) Each CustomerID, phone number, drivers licence, and email must be unique for each CUSTOMER record

b) Each ReviewID must be unique for each REVIEW record

c) Each VehicleID and Plate must be unique for each VEHICLE record

d) Each ReservationID must be unique for each RESERVATION record

e) Each RentalID must be unique for each RENTAL record

f) Each PaymentID must be unique for each PAYMENT record

g) Each MaintenanceID must be unique for each VEHICLE record

Domain Constraints:

a) The value of PhoneNumber in a CUSTOMER record must be a valid phone number (000-000-0000)

b) The value of Email in a CUSTOMER record must be a valid email address (something@domain.com)

c) The value of Type in a VEHICLE record must be one of the values in the set {Economy, Luxury}

d) The value of DailyRate and FuelCapacity in a VEHICLE record must be a positive number

e) The value of Status in a VEHICLE record must be one of the values in the set {Available, Rented, Maintenance}

f) The value of StartDate and EndDate in a RESERVATION record must be a valid date (YYY-MM-DD)

g) The value of ReservationStatus in a RESERVATION record must be one of the values in the set {Cancelled, Confirmed, Pending}

h) The value of PickupDate and ReturnDate in a RENTAL record must be a valid date (YYY-MM-DD)

i) The value of Rating in a REVIEW record must be between 1 and 5 (inclusive)

j) The value of TotalCost in a RENTAL record must be a positive number

k) The value of Amount in a PAYMENT record must be a positive number

l) The value of PaymentDate in a PAYMENT record must be a valid date (YYY-MM-DD)

m) The value of Maintenance ID in a VEHICLE record can be NULL, as some vehicles will not have gone under maintenance.

n) The value of Status in a MAINTENANCE record must be one of the values in the set {Needs_Maintenance, Under_Maintenance, Maintenance_Completed}

Referential Integrity Constraints:

a) A value of CustomerID in a RESERVATION and REVIEW record must also exist in some CUSTOMER record

b) A value of VehicleID in a RESERVATION and REVIEW record must also exist in some VEHICLE record

c) A value of ReservationID in a RENTAL record must also exist in some RESERVATION record

d) A value of RentalID in a PAYMENT record must also exist in some RENTAL record

e) A value of MaintenanceID in a VEHICLE record must also exist in some MAINTENANCE record

Entity Integrity Constraints:

a) Every record in CUSTOMER must have a value for CustomerID

b) Every record in REVIEW ,must have a value for ReviewID

c) Every record in VEHICLE must have a value for VehicleID

d) Every record in RESERVATION must have a value for ReservationID

e) Every record in RENTAL must have a value for RentalID

f) Every record in PAYMENT must have a value for PaymentID

g) Every record in MAINTENANCE must have a value for MaintenanceID

General Semantic Integrity Constraints:

a) A VEHICLE record cannot have two reservations that overlap in time unless the earlier reservation has already ended before the later reservation begins.

b) A RESERVATION record cannot have an EndDate earlier than its StartDate

c) A RENTAL record cannot have an ReturnDate earlier than its PickupDate

Example Queries

a) List all available vehicles

b) Retrieve a reservation of a customer whose CustomerID = 102

c) List all the rentals where pickup date is after Aug 1, 2025

Example Updates

a) Add a new customer to the database whose CustomerID = 120, Name = John Doe, Address = 350 Victoria St. PhoneNumber = 123-456-7890, Email = johndoe@example.com

b) Update the status vehicle whose VehicleID = 510 to rented

Users

- Customers: Main use is to create new reservations, pay, and view availability and history. Not given direct access to the DBMS, but rather as an end user, through the form of some application interface.
  - Add/Cancel a reservation
  - View available vehicles and reservation dates
  - Pay for rentals
  - View rental history
- Operations/Maintenance Office User: Main application is to enter newly available vehicles into the database and remove out of service ones.
  - Add new vehicles available to rent
  - Remove damaged vehicles from inventory

    - Enter the status of vehicles in service or completed
- Reservation Office User: Able to enter data on bookings of vehicles, and later be able to update the status of rentals.
    - Reserve a vehicle for a customer
    - Verify a customer has successfully made their payment for their reservation
    - Add a customer to the rental
    - Check customer has a valid driver's licence
- Accounting Office User: Able to generate reports, manage financial information, verify payments, and receipts on rentals.
    - Check if all payments have been received
    - Print invoices for customers

Conclusion

The Car Rental DBMS will store and organize data efficiently, allowing users to access key information about customers, vehicles, reservations, rentals, and payments. The system is expected to aid in decision making by providing insights to vehicle availability, customer rental histories, and payment tracking. In the end, this will assist in optimizing operations and enhancing customer experience.