DBMS PROJECT REPORT

BACHELOR OF TECHNOLOGY



SCHOOL OF COMPUTING SCIENCE AND ENGINEERING GALGOTIAS UNIVERSITY, GREATER NOIDA

UTTAR PRADESH

BY:

ATULENDRA SINGH YADAV – 23SCSE1410037

ABHAY PRATAP SINGH YADAV – 23SCSE1410087

AYUSH RAJ – 23SCSE1410076

**Car Rental Management System**

### **Abstract**:

The Car Rental Management System is a database-driven project designed to automate and optimize car rental operations. It provides a structured and efficient way to manage car inventory, rental transactions, and customer details. The system ensures real-time updates, minimizes manual errors, and supports seamless management of business processes.

### **1. Objective**

The objective of this project is to create a robust database management system to handle the following:

1. **Car Inventory**: Track the availability, details, and rental pricing of cars.
2. **Customer Details**: Store and retrieve customer data for better service.
3. **Rental Transactions**: Manage car rentals, calculate costs, and monitor rental history.

The system ensures data integrity, accuracy, and ease of access, benefiting both customers and administrators.

### **2. Features of the System**

#### **A. Car Inventory Management**

* Add and update car details such as car type, name, and rental price per day.
* Monitor the availability status of each car.

#### **B. Customer Management**

* Record customer information including name, address, phone number, and email.
* Link customer IDs with rental records for efficient tracking.

#### **C. Rental Transaction Management**

* Track rental start and end dates.
* Automatically calculate the total rental cost based on the duration of the rental.
* Update car availability status upon rental or return.

#### **D. Data Integrity**

* Relationships between cars, customers, and rentals are maintained using foreign keys.
* Automated validation ensures consistency and prevents errors.

### **3. Database Design**

#### **A. ER Diagram**

The database is designed with three main entities:

1. **Cars**: Represents the inventory of vehicles available for rent.
2. **Customers**: Represents individuals or organizations renting the cars.
3. **Rentals**: Represents transactions linking customers to cars.

**Relationships**:

* One-to-Many: A customer can have multiple rentals.
* One-to-One: A car can only be rented out to one customer at a time.

#### **B. Schema Design**

1. **Cars Table**

| **Column** | **Data Type** | **Constraints** |
| --- | --- | --- |
| CarID | INT | PRIMARY KEY |
| CarName | VARCHAR(50) | NOT NULL |
| CarType | VARCHAR(50) | NOT NULL |
| RentalPricePerDay | DECIMAL(10,2) | NOT NULL |
| Availability | BOOLEAN | DEFAULT TRUE |

1. **Customers Table**

| **Column** | **Data Type** | **Constraints** |
| --- | --- | --- |
| CustomerID | INT | PRIMARY KEY |
| Name | VARCHAR(50) | NOT NULL |
| Address | TEXT |  |
| PhoneNumber | VARCHAR(15) | UNIQUE |
| Email | VARCHAR(50) | UNIQUE |

1. **Rentals Table**

| **Column** | **Data Type** | **Constraints** |
| --- | --- | --- |
| RentalID | INT | PRIMARY KEY |
| CustomerID | INT | FOREIGN KEY REFERENCES Customers(CustomerID) |
| CarID | INT | FOREIGN KEY REFERENCES Cars(CarID) |
| StartDate | DATE | NOT NULL |
| EndDate | DATE | NOT NULL |
| TotalCost | DECIMAL(10,2) |  |

### **4. SQL Queries**

#### **A. Create Tables**

sql

CopyEdit

CREATE TABLE Cars (

CarID INT PRIMARY KEY,

CarName VARCHAR(50),

CarType VARCHAR(50),

RentalPricePerDay DECIMAL(10,2),

Availability BOOLEAN DEFAULT TRUE

);

CREATE TABLE Customers (

CustomerID INT PRIMARY KEY,

Name VARCHAR(50),

Address TEXT,

PhoneNumber VARCHAR(15) UNIQUE,

Email VARCHAR(50) UNIQUE

);

CREATE TABLE Rentals (

RentalID INT PRIMARY KEY,

CustomerID INT,

CarID INT,

StartDate DATE,

EndDate DATE,

TotalCost DECIMAL(10,2),

FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID),

FOREIGN KEY (CarID) REFERENCES Cars(CarID)

);

#### **B. Insert Sample Data**

1. **Add Cars**:

sql

CopyEdit

INSERT INTO Cars (CarID, CarName, CarType, RentalPricePerDay)

VALUES (1, 'Toyota Corolla', 'Sedan', 50.00),

(2, 'Honda Civic', 'Sedan', 60.00),

(3, 'Ford Explorer', 'SUV', 100.00);

1. **Add Customers**:

sql

CopyEdit

INSERT INTO Customers (CustomerID, Name, Address, PhoneNumber, Email)

VALUES (101, 'John Doe', '123 Main St', '555-1234', 'johndoe@example.com'),

(102, 'Jane Smith', '456 Elm St', '555-5678', 'janesmith@example.com');

#### **C. Rent a Car**

sql

CopyEdit

INSERT INTO Rentals (RentalID, CustomerID, CarID, StartDate, EndDate, TotalCost)

VALUES (1, 101, 1, '2025-01-01', '2025-01-05', 200.00);

UPDATE Cars SET Availability = FALSE WHERE CarID = 1;

#### **D. Return a Car**

sql

CopyEdit

DELETE FROM Rentals WHERE RentalID = 1;

UPDATE Cars SET Availability = TRUE WHERE CarID = 1;

### **5. Implementation Steps**

1. Install a DBMS (e.g., MySQL or PostgreSQL).
2. Create the database and tables using the provided SQL queries.
3. Populate tables with sample data.
4. Execute queries to perform basic operations (e.g., renting and returning cars).
5. Test the database to ensure functionality and integrity.

### **6. Advantages**

1. **Efficiency**: Automates car rental processes, reducing manual workload.
2. **Data Integrity**: Ensures accuracy and consistency through foreign keys and constraints.
3. **Scalability**: Can easily accommodate more customers, cars, and rental transactions.
4. **Real-Time Updates**: Reflects changes in availability instantly.

### **7. Future Enhancements**

* Integrate a user-friendly web or mobile interface.
* Add analytics features to track usage trends and optimize inventory.
* Include online payment integration for seamless transactions.
* Implement a recommendation system to suggest cars based on user preferences.

### **8. Conclusion**

The Car Rental Management System is a practical solution for businesses to streamline operations, improve customer satisfaction, and maintain accurate records. By leveraging database management principles, the system ensures reliability, scalability, and ease of use.