I.T.VEDANT

Decode your dreams

Name : Rigved Jagdish Tupsakhare

Project Name : Car Rental Management System

Project Guide : Prof. Sameer Warsolkar

Car Rental Management System

Car Rental Management System

1.Objective

The objective of the Car Rental Management System is to create a comprehensive database and management system for a car rental business. The system aims to streamline the processes involved in managing cars, customers, rentals, and payments, providing efficient tracking and reporting capabilities.

Here are the key operations that the system can perform:

1. Car Operations:

Add Car: Add new cars to the rental fleet, specifying details such as brand, model, year, and rental fees per day.

Update Car Information: Modify details of existing cars, such as rental fees or other specifications.

2. Customer Operations:

Add Customer: Add new customers to the system, including their names, contact details, and email addresses.

Update Customer Information: Modify customer details as needed for accuracy.

Retrieve Customer Details: Retrieve customer information for rental transactions and customer service.

3. Rental Operations:

Initiate Rental: Start a new rental transaction, associating a specific car with a customer and recording the rental start date.

Update Rental Details: Modify rental details, such as rental end date or associated car or customer information.

Complete Rental: Mark a rental as completed when the car is returned, updating rental end date and status.

Cancel Rental: Cancel a rental if needed, updating the rental status accordingly.

4. Payment Operations:

Record Payment: Record payments associated with each rental transaction.

Calculate Total Amount Paid: Calculate and display the total amount paid for each rental.

Identify Overdue Payments: Identify and manage overdue payments, updating payment history.

5. Reporting and Insights Operations:

Generate Reports: Create reports to gather insights into various aspects of the business, such as car availability, overdue rentals, and revenue generation.

Make Informed Decisions: Use data-driven reports to make informed decisions about business operations and strategies.

6. User Interface Operations:

Data Entry: Provide an interface for users to input new data, update existing records, and manage the database.

Retrieve Information: Allow users to retrieve information easily, facilitating efficient operations.

7. Scalability Operations:

Add More Cars and Customers: Accommodate the growth of the business by adding more cars and customers to the system.

Handle Increased Rentals: Manage an increasing number of rental transactions efficiently.

2.ER Diagram for Car Rental Management System

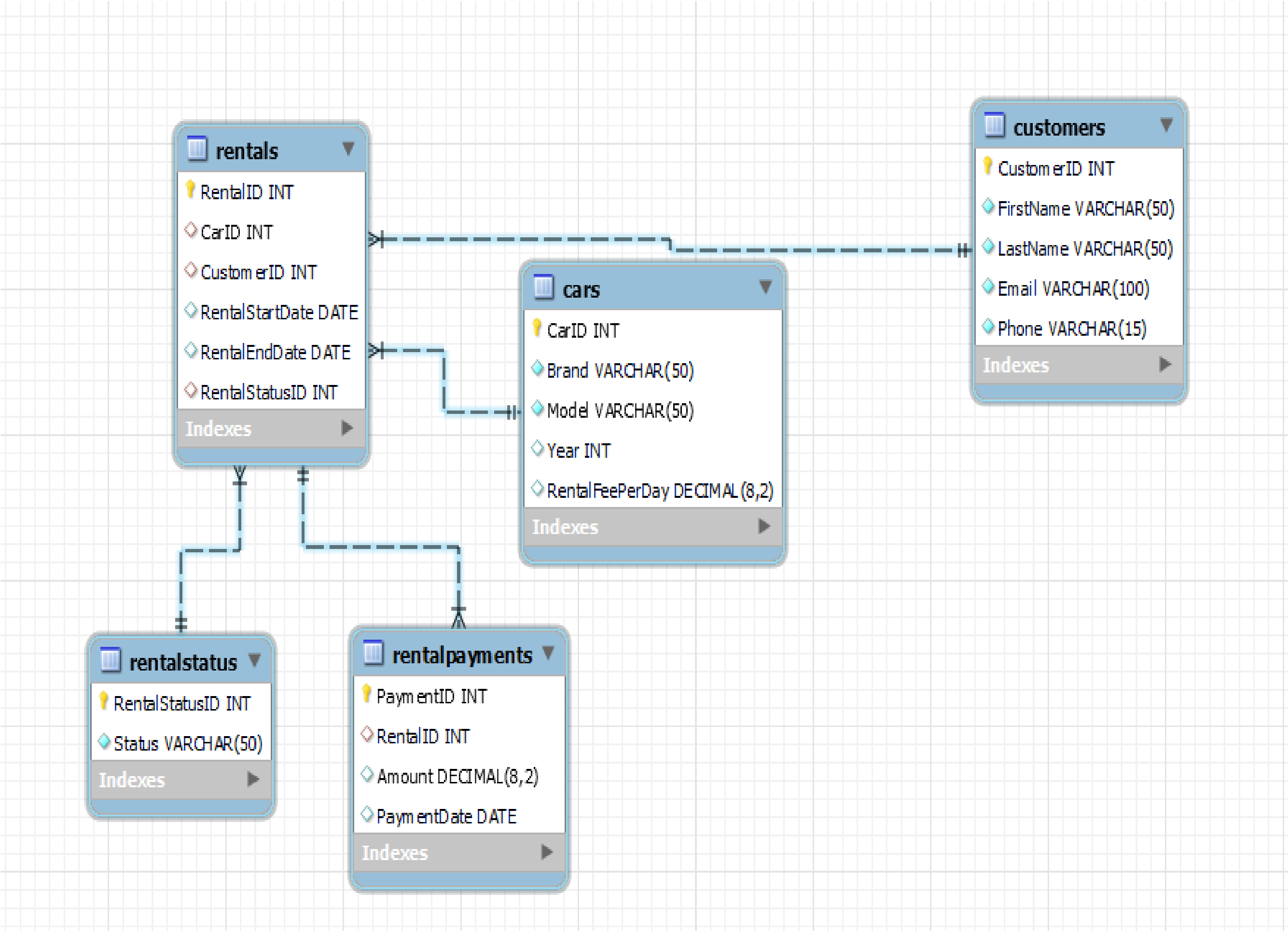
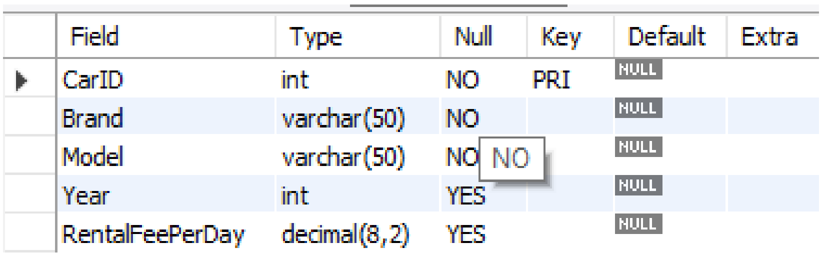


Table Names:

1. Cars
2. Customers
3. Rentals
4. Rental Status
5. Rental Payments

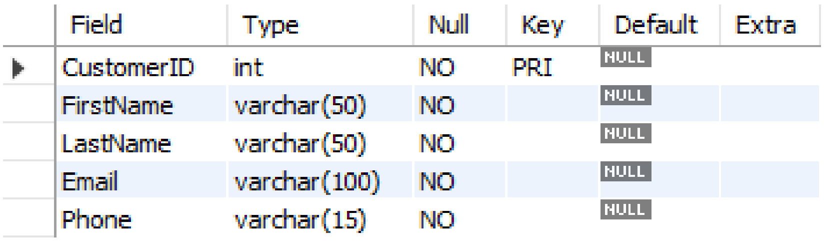
3.Table Description

1.Cars:



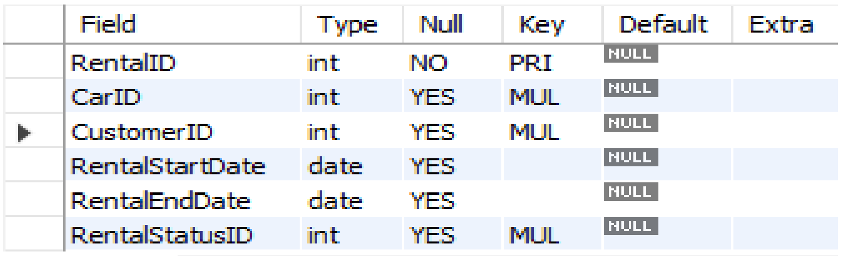
Field	Type	Null	Key	Default	Extra
CarID	int	NO	PRI	NULL	
Brand	varchar(50)	NO		NULL	
Model	varchar(50)	NO		NULL	
Year	int	YES		NULL	
RentalFeePerDay	decimal(8,2)	YES		NULL	

2.Customers:



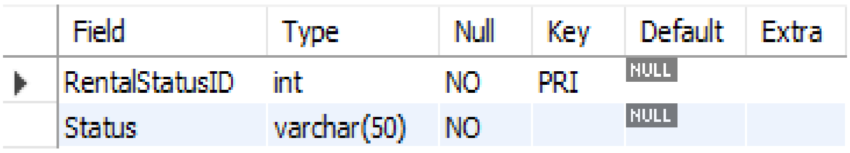
Field	Type	Null	Key	Default	Extra
CustomerID	int	NO	PRI	NULL	
FirstName	varchar(50)	NO		NULL	
LastName	varchar(50)	NO		NULL	
Email	varchar(100)	NO		NULL	
Phone	varchar(15)	NO		NULL	

3.Rentals:



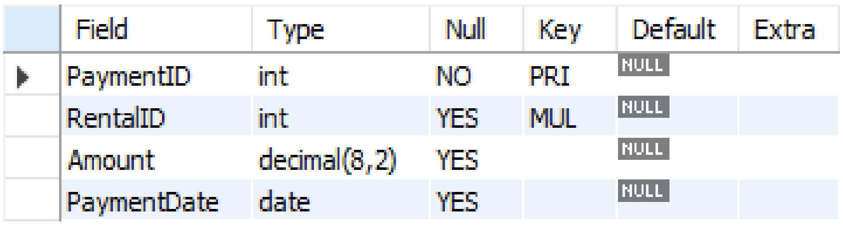
Field	Type	Null	Key	Default	Extra
RentalID	int	NO	PRI	NULL	
CarID	int	YES	MUL	NULL	
CustomerID	int	YES	MUL	NULL	
RentalStartDate	date	YES		NULL	
RentalEndDate	date	YES		NULL	
RentalStatusID	int	YES	MUL	NULL	

4. Rental Status:



Field	Type	Null	Key	Default	Extra
RentalStatusID	int	NO	PRI	NULL	
Status	varchar(50)	NO		NULL	

5.Rental Payments:



Field	Type	Null	Key	Default	Extra
PaymentID	int	NO	PRI	NULL	
RentalID	int	YES	MUL	NULL	
Amount	decimal(8,2)	YES		NULL	
PaymentDate	date	YES		NULL	

4.Commands

Create Commands:

Create database:

```
create database Car_Rental_System;

use Car_Rental_System;
```

Create Table Cars:

```
CREATE TABLE Cars (
    CarID INT PRIMARY KEY,
    Brand VARCHAR(50) NOT NULL,
    Model VARCHAR(50) NOT NULL,
    Year INT,
    RentalFeePerDay DECIMAL(8, 2));
```

Create Table Customers:

```
CREATE TABLE Customers (
    CustomerID INT PRIMARY KEY,
    FirstName VARCHAR(50) NOT NULL,
    LastName VARCHAR(50) NOT NULL,
    Email VARCHAR(100) NOT NULL,
    Phone VARCHAR(15) NOT NULL);
```

Create Table Rentals:

```
CREATE TABLE Rentals (
    RentalID INT PRIMARY KEY,
    CarID INT,
    CustomerID INT,
    RentalStartDate DATE,
    RentalEndDate DATE,
    RentalStatusID INT,
    FOREIGN KEY (CarID) REFERENCES Cars(CarID),
    FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID),
    FOREIGN KEY (RentalStatusID) REFERENCES RentalStatus(RentalStatusID));
```

Create Table RentalStatus:

```
CREATE TABLE RentalStatus (
    RentalStatusID INT PRIMARY KEY,
    Status VARCHAR(50) NOT NULL
);
```

Create Table RentalPayments:

```
CREATE TABLE RentalPayments (
    PaymentID INT PRIMARY KEY,
    RentalID INT,
    Amount DECIMAL(8, 2),
    PaymentDate DATE,
    FOREIGN KEY (RentalID) REFERENCES Rentals(RentalID)
);
```

Insert Commands:

-- Insert data into Cars table

```
INSERT INTO Cars (CarID, Brand, Model, Year, RentalFeePerDay) VALUES
(1, 'Toyota', 'Camry', 2019, 50.00),
(2, 'Honda', 'Civic', 2020, 45.00),
(3, 'Ford', 'Mustang', 2018, 80.00),
(4, 'Chevrolet', 'Malibu', 2021, 55.00),
(5, 'Nissan', 'Altima', 2017, 60.00),
(6, 'BMW', '3 Series', 2022, 90.00),
(7, 'Audi', 'A4', 2018, 85.00),
(8, 'Hyundai', 'Sonata', 2019, 55.00),
(9, 'Kia', 'Optima', 2020, 52.00),
(10, 'Mercedes', 'C-Class', 2021, 95.00),
(11, 'Volkswagen', 'Passat', 2017, 58.00),
(12, 'Subaru', 'Legacy', 2022, 65.00),
(13, 'GMC', 'Terrain', 2019, 75.00),
(14, 'Tesla', 'Model 3', 2020, 120.00),
(15, 'Jaguar', 'XF', 2018, 110.00),
(16, 'Toyota', 'Rav4', 2021, 70.00),
(17, 'Ford', 'Explorer', 2019, 85.00),
(18, 'Chevrolet', 'Equinox', 2020, 65.00),
(19, 'Honda', 'Pilot', 2018, 90.00),
(20, 'Nissan', 'Rogue', 2022, 75.00),
(21, 'Mercedes', 'GLE', 2021, 100.00),
(22, 'Audi', 'Q5', 2017, 80.00),
(23, 'Hyundai', 'Tucson', 2023, 60.00),
(24, 'Kia', 'Sorento', 2020, 78.00),
(25, 'Subaru', 'Outback', 2022, 68.00);

-- Insert data into Customers table
INSERT INTO Customers (CustomerID, FirstName, LastName, Email, Phone) VALUES
(101, 'John', 'Doe', 'john.doe@email.com', '555-1234'),
(102, 'Alice', 'Smith', 'alice@email.com', '555-5678'),
(103, 'Bob', 'Johnson', 'bob@email.com', '555-9876'),
(104, 'Emma', 'White', 'emma@email.com', '555-4321'),
(105, 'Chris', 'Brown', 'chris@email.com', '555-8765'),
(106, 'Olivia', 'Davis', 'olivia@email.com', '555-1111'),
(107, 'Daniel', 'Miller', 'daniel@email.com', '555-2222'),

(108, 'Sophia', 'Anderson', 'sophia@email.com', '555-3333'),

(109, 'Jackson', 'Harris', 'jackson@email.com', '555-4444'),

(110, 'Aria', 'Wilson', 'aria@email.com', '555-5555'),

(111, 'Ethan', 'Moore', 'ethan@email.com', '555-6666'),

(112, 'Mia', 'Clark', 'mia@email.com', '555-7777'),

(113, 'Liam', 'Taylor', 'liam@email.com', '555-8888'),

(114, 'Ava', 'Martin', 'ava@email.com', '555-9999'),

(115, 'Noah', 'Wright', 'noah@email.com', '555-0000');

-- Insert data into Rentals table

INSERT INTO Rentals (RentalID, CarID, CustomerID, RentalStartDate, RentalEndDate, RentalStatusID) VALUES

(201, 1, 101, '2023-01-01', '2023-01-05', 1),

(202, 3, 102, '2023-02-01', '2023-02-10', 2),

(203, 2, 103, '2023-03-01', '2023-03-07', 1),

(204, 4, 104, '2023-04-01', '2023-04-03', 3),

(205, 5, 105, '2023-05-01', '2023-05-15', 1),

(206, 6, 106, '2023-06-01', '2023-06-10', 2),

(207, 7, 107, '2023-07-01', '2023-07-07', 1),

(208, 8, 108, '2023-08-01', '2023-08-03', 3),

(209, 9, 109, '2023-09-01', '2023-09-15', 1),

(210, 10, 110, '2023-10-01', '2023-10-10', 2),

(211, 11, 111, '2023-11-01', '2023-11-05', 1),

(212, 12, 112, '2023-12-01', '2023-12-10', 2),

(213, 13, 113, '2024-01-01', '2024-01-07', 1),

(214, 14, 114, '2024-02-01', '2024-02-03', 3),

(215, 15, 115, '2024-03-01', '2024-03-15', 1),

(216, 16, 101, '2024-04-01', '2024-04-10', 2),
(217, 17, 102, '2024-05-01', '2024-05-15', 1),
(218, 18, 103, '2024-06-01', '2024-06-10', 2),
(219, 19, 104, '2024-07-01', '2024-07-07', 1),
(220, 20, 105, '2024-08-01', '2024-08-03', 3),
(221, 21, 106, '2024-09-01', '2024-09-15', 1),
(222, 22, 107, '2024-10-01', '2024-10-10', 2),
(223, 23, 108, '2024-11-01', '2024-11-05', 1),
(224, 24, 109, '2024-12-01', '2024-12-10', 2),
(225, 25, 110, '2025-01-01', '2025-01-07', 1);
-- Insert data into RentalStatus table
INSERT INTO RentalStatus (RentalStatusID, Status) VALUES
(1, 'Active'),
(2, 'Completed'),
(3, 'Canceled');
-- Insert data into RentalPayments table
INSERT INTO RentalPayments (PaymentID, RentalID, Amount, PaymentDate) VALUES
(301, 201, 250.00, '2023-01-05'),
(302, 202, 800.00, '2023-02-10'),
(303, 203, 315.00, '2023-03-07'),
(304, 204, 120.00, '2023-04-03'),
(305, 205, 900.00, '2023-05-15'),
(306, 206, 810.00, '2023-06-10'),
(307, 207, 350.00, '2023-07-07'),
(308, 208, 130.00, '2023-08-03'),
(309, 209, 720.00, '2023-09-15'),
(310, 210, 850.00, '2023-10-10'),
(311, 211, 211, '2023-11-05'),
(312, 212, 212, '2023-12-10'),
(313, 213, 213, '2024-01-07'),
(314, 214, 214, '2024-02-03'),
(315, 215, 215, '2024-03-15'),
(316, 216, 101, '2024-04-10'),
(317, 217, 102, '2024-05-15'),
(318, 218, 103, '2024-06-10'),
(319, 219, 104, '2024-07-07'),
(320, 220, 105, '2024-08-03'),
(321, 221, 106, '2024-09-15'),
(322, 222, 107, '2024-10-10'),
(323, 223, 108, '2024-11-05'),
(324, 224, 109, '2024-12-10'),
(325, 225, 110, '2025-01-07');
```

5.SUBQUERIES

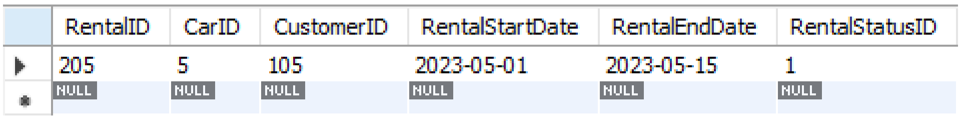
1. Subquery to Find Rentals with Maximum Payment:

SELECT *

FROM Rentals

WHERE RentalID = (SELECT RentalID FROM RentalPayments ORDER BY Amount DESC LIMIT 1);

Output:



RentalID	CarID	CustomerID	RentalStartDate	RentalEndDate	RentalStatusID
205	5	105	2023-05-01	2023-05-15	1
NULL	NULL	NULL	NULL	NULL	NULL

2. Subquery to Find Customers with Active Rentals:

SELECT *

FROM Customers

WHERE CustomerID IN (

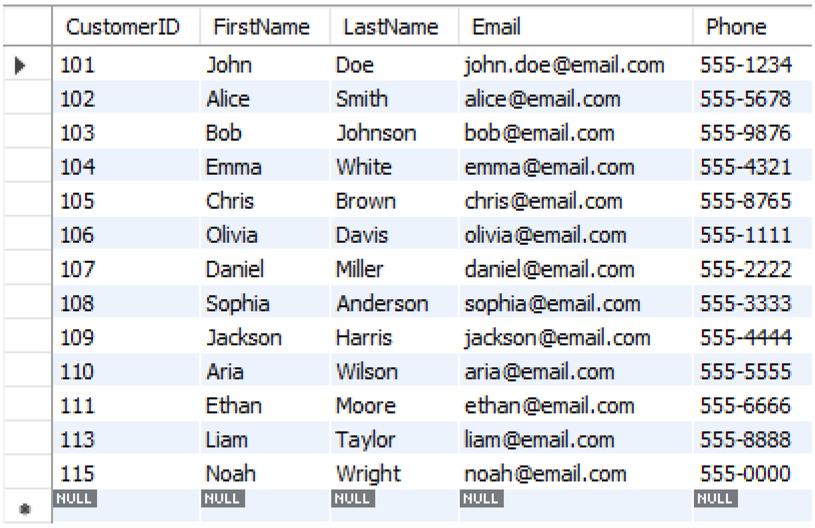
SELECT DISTINCT CustomerID

FROM Rentals

WHERE RentalStatusID = 1

);

Output:

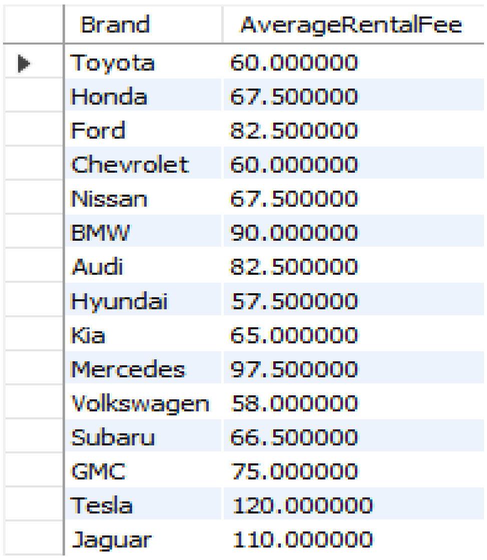


CustomerID	FirstName	LastName	Email	Phone
101	John	Doe	john.doe@email.com	555-1234
102	Alice	Smith	alice@email.com	555-5678
103	Bob	Johnson	bob@email.com	555-9876
104	Emma	White	emma@email.com	555-4321
105	Chris	Brown	chris@email.com	555-8765
106	Olivia	Davis	olivia@email.com	555-1111
107	Daniel	Miller	daniel@email.com	555-2222
108	Sophia	Anderson	sophia@email.com	555-3333
109	Jackson	Harris	jackson@email.com	555-4444
110	Aria	Wilson	aria@email.com	555-5555
111	Ethan	Moore	ethan@email.com	555-6666
113	Liam	Taylor	liam@email.com	555-8888
115	Noah	Wright	noah@email.com	555-0000
NULL	NULL	NULL	NULL	NULL

3. Subquery to Find Average Rental Fee Per Brand:

```
SELECT Brand, AVG(RentalFeePerDay) AS AverageRentalFee
FROM Cars
GROUP BY Brand;
```

Output:

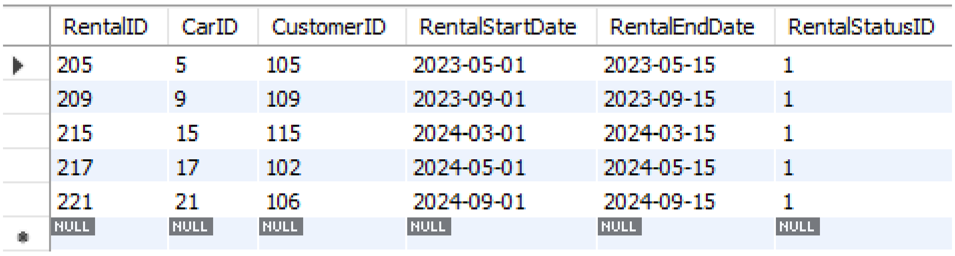


Brand	AverageRentalFee
Toyota	60.000000
Honda	67.500000
Ford	82.500000
Chevrolet	60.000000
Nissan	67.500000
BMW	90.000000
Audi	82.500000
Hyundai	57.500000
Kia	65.000000
Mercedes	97.500000
Volkswagen	58.000000
Subaru	66.500000
GMC	75.000000
Tesla	120.000000
Jaguar	110.000000

4. Subquery to Find Rentals with Longest Duration:

```
SELECT *
FROM Rentals
WHERE datediff(RentalEndDate,RentalStartDate) =(
SELECT MAX(datediff(RentalEndDate,RentalStartDate))
FROM Rentals
);
```

Output:

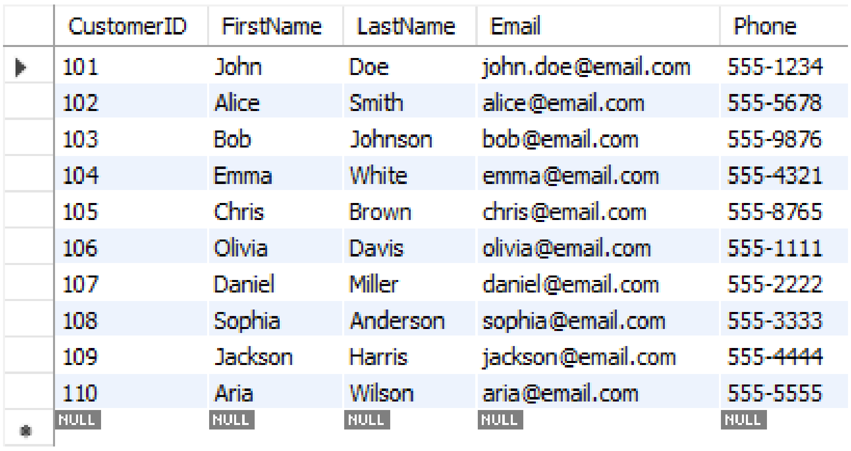


RentalID	CarID	CustomerID	RentalStartDate	RentalEndDate	RentalStatusID
205	5	105	2023-05-01	2023-05-15	1
209	9	109	2023-09-01	2023-09-15	1
215	15	115	2024-03-01	2024-03-15	1
217	17	102	2024-05-01	2024-05-15	1
221	21	106	2024-09-01	2024-09-15	1
NULL	NULL	NULL	NULL	NULL	NULL

5. Subquery to Find Customers with Multiple Rentals:

```
SELECT *
FROM Customers
WHERE CustomerID IN (
SELECT CustomerID
FROM Rentals
GROUP BY CustomerID
HAVING COUNT(RentalID) > 1
);
```

Output:



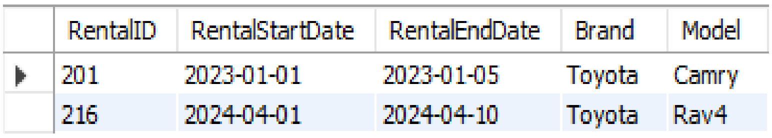
CustomerID	FirstName	LastName	Email	Phone
101	John	Doe	john.doe@email.com	555-1234
102	Alice	Smith	alice@email.com	555-5678
103	Bob	Johnson	bob@email.com	555-9876
104	Emma	White	emma@email.com	555-4321
105	Chris	Brown	chris@email.com	555-8765
106	Olivia	Davis	olivia@email.com	555-1111
107	Daniel	Miller	daniel@email.com	555-2222
108	Sophia	Anderson	sophia@email.com	555-3333
109	Jackson	Harris	jackson@email.com	555-4444
110	Aria	Wilson	aria@email.com	555-5555
NULL	NULL	NULL	NULL	NULL

6.JOINS

1. Join to Find Rentals with Cars of a Specific Brand:

```
SELECT Rentals.RentalID, Rentals.RentalStartDate, Rentals.RentalEndDate,

Cars.Brand, Cars.Model

FROM Rentals

JOIN Cars ON Rentals.CarID = Cars.CarID

WHERE Cars.Brand = 'Toyota';
```

Output:

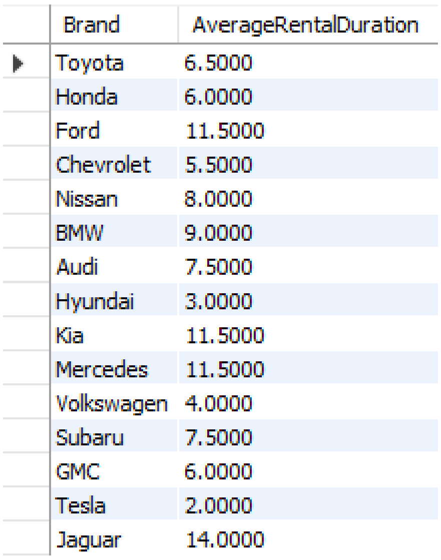


RentalID	RentalStartDate	RentalEndDate	Brand	Model
201	2023-01-01	2023-01-05	Toyota	Camry
216	2024-04-01	2024-04-10	Toyota	Rav4

2. Join to Find Average Rental Duration Per Car Brand:

```
SELECT Cars.Brand, AVG(datediff(Rentals.RentalEndDate,Rentals.RentalStartDate)) AS
AverageRentalDuration

FROM Cars

LEFT JOIN Rentals ON Cars.CarID = Rentals.CarID

GROUP BY Cars.Brand;
```

Output:

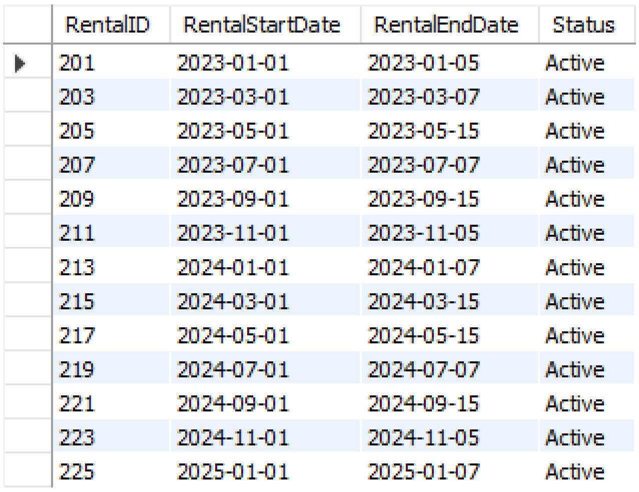


Brand	AverageRentalDuration
Toyota	6.5000
Honda	6.0000
Ford	11.5000
Chevrolet	5.5000
Nissan	8.0000
BMW	9.0000
Audi	7.5000
Hyundai	3.0000
Kia	11.5000
Mercedes	11.5000
Volkswagen	4.0000
Subaru	7.5000
GMC	6.0000
Tesla	2.0000
Jaguar	14.0000

3. Join to Get Active Rentals with Rental Status Information:

```
SELECT Rentals.RentalID, Rentals.RentalStartDate, Rentals.RentalEndDate,
RentalStatus.Status
FROM Rentals
JOIN RentalStatus ON Rentals.RentalStatusID = RentalStatus.RentalStatusID
WHERE RentalStatus.Status = 'Active';
```

Output:

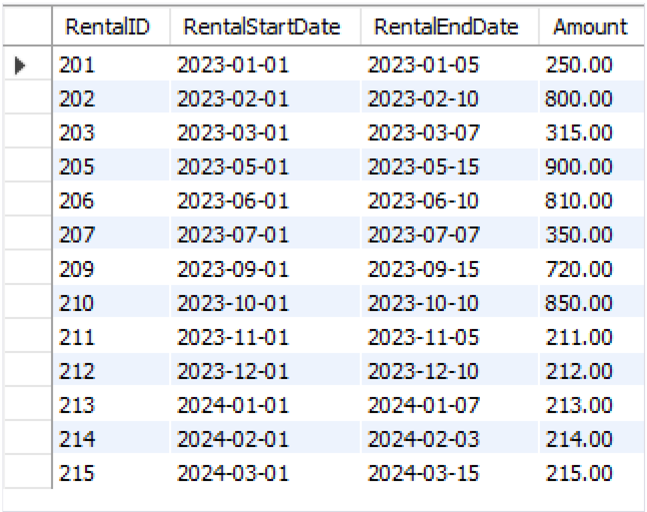


	RentalID	RentalStartDate	RentalEndDate	Status
▶	201	2023-01-01	2023-01-05	Active
	203	2023-03-01	2023-03-07	Active
	205	2023-05-01	2023-05-15	Active
	207	2023-07-01	2023-07-07	Active
	209	2023-09-01	2023-09-15	Active
	211	2023-11-01	2023-11-05	Active
	213	2024-01-01	2024-01-07	Active
	215	2024-03-01	2024-03-15	Active
	217	2024-05-01	2024-05-15	Active
	219	2024-07-01	2024-07-07	Active
	221	2024-09-01	2024-09-15	Active
	223	2024-11-01	2024-11-05	Active
	225	2025-01-01	2025-01-07	Active

4. Join to Find Rentals with Payments Exceeding a Certain Amount:

```
SELECT Rentals.RentalID, Rentals.RentalStartDate, Rentals.RentalEndDate,
RentalPayments.Amount
FROM Rentals
JOIN RentalPayments ON Rentals.RentalID = RentalPayments.RentalID
WHERE RentalPayments.Amount > 200;
```

Output:

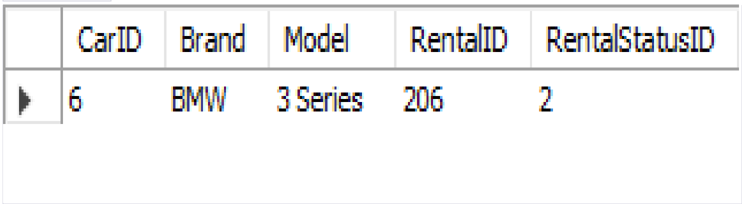


	RentalID	RentalStartDate	RentalEndDate	Amount
▶	201	2023-01-01	2023-01-05	250.00
	202	2023-02-01	2023-02-10	800.00
	203	2023-03-01	2023-03-07	315.00
	205	2023-05-01	2023-05-15	900.00
	206	2023-06-01	2023-06-10	810.00
	207	2023-07-01	2023-07-07	350.00
	209	2023-09-01	2023-09-15	720.00
	210	2023-10-01	2023-10-10	850.00
	211	2023-11-01	2023-11-05	211.00
	212	2023-12-01	2023-12-10	212.00
	213	2024-01-01	2024-01-07	213.00
	214	2024-02-01	2024-02-03	214.00
	215	2024-03-01	2024-03-15	215.00

5. Join to Find Car’s Id, model ,rental status whose brand name starts with B

```
SELECT Cars.CarID, Cars.Brand, Cars.Model,RentalID,RentalStatusID
FROM Cars
LEFT JOIN Rentals ON Cars.CarID = Rentals.CarID
WHERE Brand like "B%";
```

Output:



	CarID	Brand	Model	RentalID	RentalStatusID
▶	6	BMW	3 Series	206	2

7.CONCLUSION

In conclusion, the Car Rental Management System project has successfully addressed the key objectives of streamlining and enhancing the operations of a car rental business. The implemented system provides a robust platform for managing cars, customers, rentals, and payments, contributing to improved efficiency and decision-making within the organization.

THANK YOU!