Coding Challenge - Car Rental System – SQL

```
CREATE DATABASE car_rental_system;
USE car_rental_system;
```

-- Creating vehicle table

```
CREATE TABLE Vehicle(
vehicleID INT PRIMARY KEY,
make VARCHAR(50) NOT NULL,
model VARCHAR(50) NOT NULL,
year INT NOT NULL CHECK (year >= 2000),
dailyRate DECIMAL(10,2) NOT NULL CHECK (dailyRate > 0),
status INT CHECK (status=0 OR status=1) NOT NULL,
passengerCapacity INT NOT NULL CHECK (passengerCapacity > 0),
engineCapacity INT NOT NULL CHECK (engineCapacity > 0)
);
```

-- Creating Customer Table

```
CREATE TABLE Customer (
customerID INT PRIMARY KEY ,
firstName VARCHAR(50) NOT NULL,
lastName VARCHAR(50) NOT NULL,
email VARCHAR(100) UNIQUE NOT NULL,
phoneNumber VARCHAR(15) UNIQUE NOT NULL
);
```

-- Creating Lease Table

```
CREATE TABLE Lease (
leaseID INT PRIMARY KEY ,
vehicleID INT NOT NULL,
customerID INT NOT NULL,
startDate DATE NOT NULL,
endDate DATE NOT NULL ,
type ENUM('DailyLease', 'MonthlyLease') NOT NULL,
FOREIGN KEY (vehicleID) REFERENCES Vehicle(vehicleID) ON DELETE CASCADE,
FOREIGN KEY (customerID) REFERENCES Customer(customerID) ON DELETE CASCADE
);
```

-- Creating Payment Table

```
CREATE TABLE Payment (
paymentID INT PRIMARY KEY ,
leaseID INT NOT NULL,
paymentDate DATE NOT NULL ,
amount DECIMAL(10,2) NOT NULL CHECK (amount > 0),
FOREIGN KEY (leaseID) REFERENCES Lease(leaseID) ON DELETE CASCADE
);
```

-- Inserting values in Vehicle table

```
INSERT INTO Vehicle (vehicleID, make, model, year, dailyRate, status, passengerCapacity,
engineCapacity)
VALUES
(1, 'Toyota', 'Camry', 2022, 50.00, 1, 4, 1450),
(2, 'Honda', 'Civic', 2023, 45.00, 1, 7, 1500),
(3, 'Ford', 'Focus', 2022, 48.00, 0, 4, 1400),
(4, 'Nissan', 'Altima', 2023, 52.00, 1, 7, 1200),
(5, 'Chevrolet', 'Malibu', 2022, 47.00, 1, 4, 1800),
(6, 'Hyundai', 'Sonata', 2023, 49.00, 0, 7, 1400),
(7, 'BMW', '3 Series', 2023, 60.00, 1, 7, 2499),
(8, 'Mercedes', 'C-Class', 2022, 58.00, 1, 8, 2599),
(9, 'Audi', 'A4', 2022, 55.00, 0, 4, 2500),
(10, 'Lexus', 'ES', 2023, 54.00, 1, 4, 2500);
```

-- Inserting values in Customer table

```
INSERT INTO Customer (customerID, firstName, lastName, email, phoneNumber)
VALUES
(1, 'John', 'Doe', 'johndoe@example.com', '555-555-5555'),
(2, 'Jane', 'Smith', 'janesmith@example.com', '555-123-4567'),
(3, 'Robert', 'Johnson', 'robert@example.com', '555-789-1234'),
(4, 'Sarah', 'Brown', 'sarah@example.com', '555-456-7890'),
(5, 'David', 'Lee', 'david@example.com', '555-987-6543'),
(6, 'Laura', 'Hall', 'laura@example.com', '555-234-5678'),
(7, 'Michael', 'Davis', 'michael@example.com', '555-876-5432'),
(8, 'Emma', 'Wilson', 'emma@example.com', '555-432-1098'),
(9, 'William', 'Taylor', 'william@example.com', '555-321-6547'),
(10, 'Olivia', 'Adams', 'olivia@example.com', '555-765-4321');
```

-- Inserting values in Lease table

```
INSERT INTO Lease (leaseID, vehicleID, customerID, startDate, endDate, type)
VALUES
```

```
(1, 1, 1, '2023-01-01', '2023-01-05', 'DailyLease'),
(2, 2, 2, '2023-02-15', '2023-02-28', 'MonthlyLease'),
(3, 3, 3, '2023-03-10', '2023-03-15', 'DailyLease'),
(4, 4, 4, '2023-04-20', '2023-04-30', 'MonthlyLease'),
(5, 5, 5, '2023-05-05', '2023-05-10', 'DailyLease'),
(6, 4, 3, '2023-06-15', '2023-06-30', 'MonthlyLease'),
(7, 7, 7, '2023-07-01', '2023-07-10', 'DailyLease'),
(8, 8, 8, '2023-08-12', '2023-08-15', 'MonthlyLease'),
(9, 3, 3, '2023-09-07', '2023-09-10', 'DailyLease'),
(10, 10, 10, '2023-10-10', '2023-10-31', 'MonthlyLease');
```

-- Inserting values in Payment table

```
INSERT INTO Payment (paymentID, leaseID, paymentDate, amount)
VALUES
(1, 1, '2023-01-03', 200.00),
(2, 2, '2023-02-20', 1000.00),
(3, 3, '2023-03-12', 75.00),
(4, 4, '2023-04-25', 900.00),
(5, 5, '2023-05-07', 60.00),
(6, 6, '2023-06-18', 1200.00),
(7, 7, '2023-07-03', 40.00),
(8, 8, '2023-08-14', 1100.00),
(9, 9, '2023-09-09', 80.00),
(10, 10, '2023-10-25', 1500.00);
```

-- 1. Update the daily rate for a Mercedes car to 68.

```
UPDATE Vehicle
SET dailyRate = 68.00
WHERE vehicleID=8;
```

-- 2. Delete a specific customer and all associated leases and payments.

```
DELETE FROM Customer WHERE customerID = 2;
```

-- 3. Rename the "paymentDate" column in the Payment table to "transactionDate".

```
ALTER TABLE Payment
RENAME COLUMN paymentDate TO transactionDate;
```

-- 4. Find a specific customer by email.

```
SELECT * FROM Customer
WHERE email = 'robert@example.com';
```

-- 5.Get active leases for a specific customer.

```
SELECT endDate AS ACTIVE_LEASES FROM Lease
WHERE customerID = 2 ;
```

-- 6. Find all payments made by a customer with a specific phone number.

```
SELECT p.* FROM Payment p
JOIN Lease l ON p.leaseID = l.leaseID
JOIN Customer c ON l.customerID = c.customerID
WHERE c.phoneNumber = '555-123-4567';
```

-- 7.Calculate the average daily rate of all available cars.

```
 SELECT AVG(dailyRate) AS averageDailyRate
FROM Vehicle
```

```
WHERE status = 1;
```

-- 8. Find the car with the highest daily rate.

```
SELECT MAX(dailyRate) AS highestDailyRate
FROM Vehicle ;
```

-- 9. Retrieve all cars leased by a specific customer.

```
SELECT v.make FROM Vehicle v
JOIN Lease l ON v.vehicleID = l.vehicleID
WHERE l.customerID = 2;
```

-- 10. Find the details of the most recent lease.

```
SELECT *
FROM Lease
WHERE startDate = (SELECT MAX(startDate) FROM Lease);
```

-- 11. List all payments made in the year 2023.

```
SELECT * FROM Payment
WHERE YEAR(transactionDate) = 2023;
```

-- 12. Retrieve customers who have not made any payments.

```
SELECT c.* FROM Customer c
LEFT JOIN Lease l ON c.customerID = l.customerID
LEFT JOIN Payment p ON l.leaseID = p.leaseID
WHERE p.paymentID IS NULL;
```

-- 13. Retrieve Car Details and Their Total Payments.

```
SELECT v.vehicleID, v.make, v.model, SUM(p.amount) AS totalPayments
FROM Vehicle v
```

```sql
JOIN Lease l ON v.vehicleID = l.vehicleID
JOIN Payment p ON l.leaseID = p.leaseID
GROUP BY v.vehicleID, v.make, v.model;
```

```sql
-- 14. Calculate Total Payments for Each Customer.
SELECT c.customerID, c.firstName, c.lastName, SUM(p.amount) AS totalPayments
FROM Customer c
JOIN Lease l ON c.customerID = l.customerID
JOIN Payment p ON l.leaseID = p.leaseID
GROUP BY c.customerID, c.firstName, c.lastName;
```

```sql
-- 15. List Car Details for Each Lease.
SELECT l.leaseID, v.make, v.model, v.year, l.startDate, l.endDate
FROM Lease l
JOIN Vehicle v ON l.vehicleID = v.vehicleID;
```

```sql
-- 16. Retrieve Details of Active Leases with Customer and Car Information.
SELECT l.leaseID, c.firstName, c.lastName, v.make, v.model, l.startDate, l.endDate
FROM Lease l
JOIN Customer c ON l.customerID = c.customerID
JOIN Vehicle v ON l.vehicleID = v.vehicleID
WHERE l.endDate >= CURDATE();
```

```sql
-- 17. Find the Customer Who Has Spent the Most on Leases.
SELECT c.customerID, c.firstName, c.lastName, SUM(p.amount) AS totalSpent
FROM Customer c
JOIN Lease l ON c.customerID = l.customerID
JOIN Payment p ON l.leaseID = p.leaseID
GROUP BY c.customerID, c.firstName, c.lastName
```

```sql
ORDER BY totalSpent DESC
LIMIT 1;
```

-- 18. List All Cars with Their Current Lease Information.

```sql
SELECT v.vehicleID, v.make, v.model, v.year, l.startDate, l.endDate, c.firstName, c.lastName
FROM Vehicle v
LEFT JOIN Lease l ON v.vehicleID = l.vehicleID
LEFT JOIN Customer c ON l.customerID = c.customerID;
```