CAR RENTAL SYSTEM – CODING CHALLENGE

BY KARTHIKA KALIMUTHU, Excel Engineering College

```
create database CarRentalSystem
use CarRentalSystem

create table Vehicle (
                    vehicleID int primary key,
                    make varchar (30),
                    model varchar (30),
                    year int,
                    dailyRate decimal (5,2),
                    status bit,
                    passengerCapacity int,
                    engineCapacity int
                    )

create table Customer (
                    customerID int primary key,
                    firstName varchar (30),
                    lastName varchar (30),
                    email varchar (100),
                    phoneNumber int
                    )
create table Lease (
                    leaseID int primary key,
                    vehicleID int,
                    customerID int,
```

```
                startDate date,
                endDate date,
                type varchar (20),
                foreign key (vehicleID) references Vehicle (vehicleID),
                foreign key (customerID) references Customer (customerID),
                )

create table Payment (
                paymentID int primary key,
                leaseID int,
                paymentDate date,
                amount decimal (10,2)
                foreign key (leaseID) references Lease (leaseID)
                )

insert into Vehicle 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, 'Mercedes', 'C-Class', 2023, 60.00, 0, 7,2499)

select * from Vehicle

insert into Customer values
(1, 'John', 'Doe', 'johndoe@example.com', 555555555),
(2, 'Jane', 'Smith', 'janesmith@example.com', 555123456),
(3, 'Robert', 'Johnson', 'robert@example.com', 555789123),
(4, 'Sarah', 'Brown', 'sarah@example.com', 555456789),
```

(5, 'David', 'Lee', 'david@example.com', 555987654)

select * from Customer

insert into Lease values

(1, 1, 2, '2025-01-01', '2025-01-05', 'Daily'),

(2, 1, 1, '2025-02-15', '2025-02-28', 'Monthly'),

(3, 2, 4, '2025-03-10', '2025-03-15', 'Daily'),

(4, 4, 2, '2025-04-20', '2025-04-30', 'Monthly'),

(5, 3, 5, '2025-06-09', '2025-06-19', 'Daily')

select * from Lease

insert into Payment 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)

select * from Payment

----------**Queries**----------

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

update Vehicle set dailyRate = 68.00 where make = 'Mercedes'

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

```
delete from Payment
where leaseID in (select leaseID from Lease where customerID = 1)
delete from Lease where customerID = 1
delete from Customer where customerID = 1
```

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

```
select paymentDate as transactionDate from Payment
```

4. Find a specific customer by email.

```
select firstName from Customer
where email = 'sarah@example.com'
```

5. Get active leases for a specific customer.

```
select * from Lease
where customerID = 5 and endDate > getdate ()
```

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

```
select c.phoneNumber, p.paymentID, p.leaseID, p.paymentDate,p.amount from Customer as c
join Lease as l
on c.customerID = l.leaseID
join Payment as p
on l.leaseID = p.leaseID
where c.phoneNumber = 555555555
```

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

```
select status as available, avg (dailyRate) avg_daily_rt from Vehicle
where status =1
group by status
```

8. Find the car with the highest daily rate.

```
select Make as CarName, dailyRate as Max_DR from Vehicle
where dailyRate = (select max (dailyRate) from Vehicle)
```

9. Retrieve all cars leased by a specific customer.

```
select * from Vehicle as v
join Lease as l
on v.vehicleID = l.vehicleID
where l.customerID = 1
```

10. Find the details of most recent lease.

```
select top (1) * from Lease
order by startDate desc
```

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

```
select * from Payment
where year(paymentDate) = 2023
```

12. Retrieve customers who have not made any payments.

```
select * from Customer as c
where c.customerID not in (select distinct l.customerID from Lease as l
join Payment as p
on l.leaseID = p.leaseID)
```

13. Retrieve car details and their total payments.

```
select v.vehicleID, v.make, v.model, sum (p.amount) as tot_pay from Vehicle as v
join Lease as l
on v.vehicleID = l.vehicleID
join Payment as p
```

```
on l.leaseID = p.leaseID
group by v.vehicleID,v.make, v.model
```

14. Calculate total payments for each customer.

```
select c.customerID, c.firstName, c.lastName, sum (p.amount) as tot_pay from Customer as c
join Lease as l
on c.customerID = l.customerID
join Payment as p
on l.leaseID = p.leaseID
group by c.customerID, c.firstName, c.lastName
```

15. List car details for each lease.

```
select l.leaseID, v.vehicleID, v.make, v.model from Vehicle as v
join Lease as l
on v.vehicleID = l.vehicleID
```

16. Retrieve details of active leases with customer and car information.

```
select L.*,c.firstName, c.lastName, v.make, v.model from Lease as l
join Customer as c
on l.customerID = c.customerID
join Vehicle as v
on l.vehicleID = v.vehicleID
where endDate > getdate()
```

17. Find the customer who has spent the most amount on leases.

```
select top 1 c.customerID, c.firstName, c.lastName, sum (p.amount) as tot_spt from Customer as c
join Lease as l
on c.customerID = l.customerID
join Payment as p
```

```
on l.leaseID = p.leaseID

group by c.customerID, c.firstName, c.lastName
```

18. List all cars with their current lease information.

```
select v.vehicleID, v.make, v.model, l.leaseID, l.startDate, l.endDate from Vehicle as v

left join Lease as l

on v.vehicleID = l.vehicleID and l.endDate > getdate ()
```