Coding Challenge - Car Rental System – SQL

CREATE DATABASE carrentalsystem;

use carrentalsystem;

CREATE TABLE Vehicle (

vehicleID INT PRIMARY KEY,

make VARCHAR(50),

model VARCHAR(50),

year INT,

dailyRate DECIMAL(10,2),

status ENUM('available', 'notAvailable'),

passengerCapacity INT,

engineCapacity INT

);

drop table vehicle;

-- Create Vehicle Table

CREATE TABLE Vehicle (

vehicleID INT PRIMARY KEY,

make VARCHAR(50),

model VARCHAR(50),

year INT,

dailyRate DECIMAL(10,2),

status INT CHECK (status IN (0,1)), -- 0 = Not Available, 1 = Available

passengerCapacity INT,

engineCapacity INT

);

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);

select\* from Vehicle;

CREATE TABLE Customer (

customerID INT PRIMARY KEY,

firstName VARCHAR(50),

lastName VARCHAR(50),

email VARCHAR(100) UNIQUE,

phoneNumber VARCHAR(15) UNIQUE

);

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');

select\* from Customer;

CREATE TABLE Lease (

leaseID INT PRIMARY KEY,

vehicleID INT,

customerID INT,

startDate DATE,

endDate DATE,

leaseType ENUM('Daily', 'Monthly'),

FOREIGN KEY (vehicleID) REFERENCES Vehicle(vehicleID) ON DELETE CASCADE,

FOREIGN KEY (customerID) REFERENCES Customer(customerID) ON DELETE CASCADE

);

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

(1, 1, 1, '2023-01-01', '2023-01-05', 'Daily'),

(2, 2, 2, '2023-02-15', '2023-02-28', 'Monthly'),

(3, 3, 3, '2023-03-10', '2023-03-15', 'Daily'),

(4, 4, 4, '2023-04-20', '2023-04-30', 'Monthly'),

(5, 5, 5, '2023-05-05', '2023-05-10', 'Daily'),

(6, 4, 3, '2023-06-15', '2023-06-30', 'Monthly'),

(7, 7, 7, '2023-07-01', '2023-07-10', 'Daily'),

(8, 8, 8, '2023-08-12', '2023-08-15', 'Monthly'),

(9, 3, 3, '2023-09-07', '2023-09-10', 'Daily'),

(10, 10, 10, '2023-10-10', '2023-10-31', 'Monthly');

select\* from Lease;

create table Payment(

paymentID int primary key,

leaseID int,

paymentDate DATE,

amount DECIMAL(10,2),

FOREIGN KEY (leaseID) REFERENCES Lease(leaseID) ON DELETE CASCADE

);

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.

-- FIRST

UPDATE Vehicle

set dailyRate=68

where make='Mercedes';

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

-- SECOND

DELETE FROM Customer

WHERE customerID = 4;

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

-- THIRD

ALTER TABLE Payment

RENAME COLUMN paymentDate TO transactionDate;

1. Find a specific customer by email.

-- FOURTH

SELECT CONCAT(firstname,'',lastname) as fullname

from customer

where email='laura@example.com';

1. Get active leases for a specific customer.

-- fifth

SELECT \* FROM Lease

WHERE customerID = 1

AND endDate = CURRENT\_DATE;

-- for fifth ques there is no end date as current so i have changed 1 st customers end date as current

UPDATE Lease

SET endDate = CURDATE()

WHERE customerID = 1; -- the first customer has customerID = 1

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

-- 6

select p.\* from payment p

join lease l on l.leaseId =p.leaseId join customer c on c.customerId =l.customerId

where email='olivia@example.com';

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

-- 7

select avg(dailyrate) as average\_daily\_rate

from Vehicle

where status=1;

1. Find the car with the highest daily rate.

-- 8 WITH LIMIT

select \* from vehicle

order by dailyrate desc

limit 1;

-- 8 WITH SUBQUERY

SELECT \*

FROM Vehicle

WHERE dailyRate = (SELECT MAX(dailyRate) FROM Vehicle);

1. Retrieve all cars leased by a specific customer.

-- 9

SELECT V.\* FROM Vehicle V

JOIN Lease L ON V.vehicleID = L.vehicleID

WHERE L.customerID = 2;

1. Find the details of the most recent lease.

-- 10

SELECT \* from lease

order by startDate desc

limit 1;

-- subquery

SELECT \*

FROM Lease

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

1. List all payments made in the year 2023.

-- 11

SELECT \* FROM Payment

WHERE YEAR(transactionDate) = 2023;

1. Retrieve customers who have not made any payments.

-- 12.

SELECT \* FROM Customer

WHERE customerID NOT IN (SELECT DISTINCT customerID FROM Lease L

JOIN Payment P ON L.leaseID = P.leaseID);

1. Retrieve Car Details and Their Total Payments.

-- 13

SELECT V.\*, SUM(P.amount) AS totalPayments

FROM Vehicle V

JOIN Lease L ON V.vehicleID = L.vehicleID

JOIN Payment P ON L.leaseID = P.leaseID

GROUP BY V.vehicleID;

1. Calculate Total Payments for Each Customer.

-- 14.

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;

1. List Car Details for Each Lease.

-- 15

SELECT L.leaseID, L.startDate, L.endDate, L.leaseType, V.\*

FROM Lease L

JOIN Vehicle V ON L.vehicleID = V.vehicleID;

1. Retrieve Details of Active Leases with Customer and Car Information.

-- 16

SELECT L.\*, C.firstName, C.lastName, V.make, V.model

FROM Lease L

JOIN Customer C ON L.customerID = C.customerID

JOIN Vehicle V ON L.vehicleID = V.vehicleID

WHERE L.endDate = CURRENT\_DATE;

1. Find the Customer Who Has Spent the Most on Leases.

-- 17

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

ORDER BY totalSpent DESC

LIMIT 1;

1. List All Cars with Their Current Lease Information.

-- 18

SELECT v.\* ,l.\*, c.customerID,

CONCAT(c.firstName, ' ', c.lastName) AS customerName

FROM Vehicle v

JOIN Lease l ON v.vehicleID = l.vehicleID

JOIN Customer c ON l.customerID = c.customerID

WHERE l.startDate <= CURRENT\_DATE

AND l.endDate >= CURRENT\_DATE;