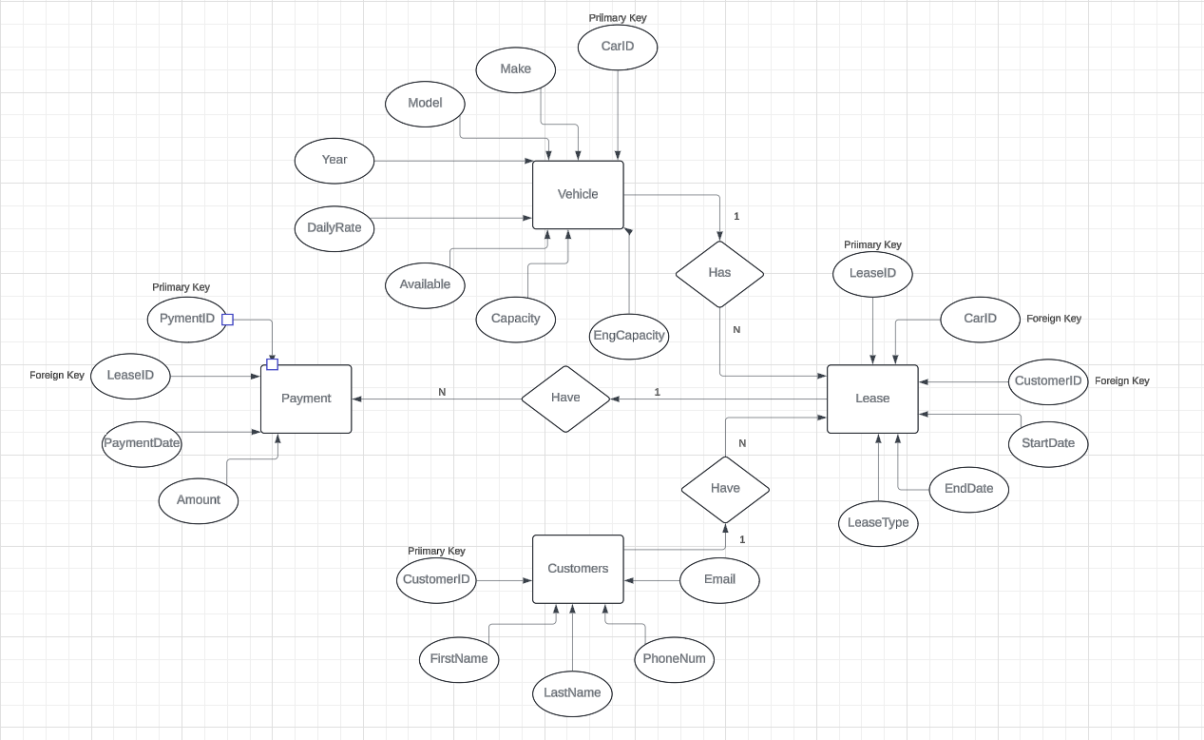
Coding Challenge - Car Rental System – SQL

J606 Anuj Jaiswal Java Batch 6

Entity Relationship Diagram:



Query Outputs

- Database ,Table Creation & Data entries

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	10:39:13	CREATE DATABASE Car_Rental_System	1 row(s) affected	0.000 sec
2	10:39:16	USE Car_Rental_System	0 row(s) affected	0.000 sec
3	10:39:19	CREATE TABLE Vehicle (carID INT PRIMARY KEY, make VARCHAR(50), model VARCHAR(50), yea...	0 row(s) affected	0.015 sec
4	10:39:21	INSERT INTO Vehicle (carID, make, model, year, dailyRate, available, passengerCapacity, engineCapacity) VAL...	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.016 sec
5	10:39:25	CREATE TABLE Customer (customerID INT PRIMARY KEY, firstName VARCHAR(50), lastName VARC...	0 row(s) affected	0.031 sec
6	10:39:27	INSERT INTO Customer (customerID, firstName, lastName, email, phoneNumber) VALUES (1, 'John', 'Doe', 'john...	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.016 sec
7	10:39:32	CREATE TABLE Lease (leaseID INT PRIMARY KEY, carID INT, customerID INT, startDate DATE, ...	0 row(s) affected	0.047 sec
8	10:39:35	INSERT INTO Lease (leaseID, carID, customerID, startDate, endDate, leaseType) VALUES (1, 1, 1, '2023-01-01...	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.000 sec
9	10:39:39	CREATE TABLE Payment (paymentID INT PRIMARY KEY, leaseID INT, paymentDate DATE, amoun...	0 row(s) affected	0.032 sec
10	10:39:43	INSERT INTO Payment (paymentID, leaseID, paymentDate, amount) VALUES (1, 1, '2023-01-03', 200.00), (2, 2, ...	3 row(s) affected Records: 3 Duplicates: 0 Warnings: 0	0.016 sec

```sql
-- 1.Update the daily rate for a Mercedes car to 68.
UPDATE Vehicle
SET dailyRate = 68.00
WHERE make = 'Mercedes';
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

carID	make	model	year	dailyRate	available	passengerCapacity	engineCapacity
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	68.00	1	8	2599
9	Audi	A4	2022	55.00	0	4	2500
10	Lexus	ES	2023	54.00	1	4	2500
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

```sql
-- 2.Delete a specific customer and all associated leases and payments.
DELETE Payment
FROM Payment
JOIN Lease ON Payment.leaseID = Lease.leaseID
WHERE Lease.customerID = 3;
DELETE FROM Lease WHERE customerID = 3;
DELETE FROM Customer WHERE customerID = 3;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

paymentID	leaseID	transactionDate	amount
1	1	2023-01-03	200.00
2	2	2023-02-20	1000.00
NULL	NULL	NULL	NULL

```sql
-- 3.Rename the "paymentDate" column in the Payment table to "transactionDate".
ALTER TABLE Payment
RENAME COLUMN paymentDate TO transactionDate;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

paymentID	leaseID	transactionDate	amount
1	1	2023-01-03	200.00
2	2	2023-02-20	1000.00
NULL	NULL	NULL	NULL

```sql
-- 4.Find a specific customer by email
SELECT * FROM Customer WHERE email = 'johndoe@example.com';
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

customerID	firstName	lastName	email	phoneNumber
1	John	Doe	johndoe@example.com	555-555-5555
NULL	NULL	NULL	NULL	NULL

```sql
-- 5.Get active leases for a specific customer.
SELECT * FROM Lease WHERE customerID = 3 AND endDate >= CURDATE();
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

leaseID	carID	customerID	startDate	endDate	leaseType
NULL	NULL	NULL	NULL	NULL	NULL

```sql
-- 6.Find all payments made by a customer with a specific phone number.
SELECT Payment.*
FROM Payment
JOIN Lease ON Payment.leaseID = Lease.leaseID
JOIN Customer ON Lease.customerID = Customer.customerID
WHERE Customer.phoneNumber = '555-555-5555';
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

paymentID	leaseID	paymentDate	amount
1	1	2023-01-03	200.00

```sql
-- 7.Calculate the average daily rate of all available cars
SELECT AVG(dailyRate) AS avgDailyRate FROM Vehicle WHERE available = 1;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

avgDailyRate
53.714286

```sql
-- 8.Find the car with the highest daily rate.
SELECT * FROM Vehicle ORDER BY dailyRate DESC LIMIT 1;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: | Fetch rows:

carID	make	model	year	dailyRate	available	passengerCapacity	engineCapacity
8	Mercedes	C-Class	2022	68.00	1	8	2599
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

```sql
-- 9.Retrieve all cars leased by a specific customer
SELECT Vehicle.*
FROM Vehicle
JOIN Lease ON Vehicle.carID = Lease.carID
WHERE Lease.customerID = 3;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

carID	make	model	year	dailyRate	available	passengerCapacity	engineCapacity
3	Ford	Focus	2022	48.00	0	4	1400
4	Nissan	Altima	2023	52.00	1	7	1200
3	Ford	Focus	2022	48.00	0	4	1400

```sql
-- 10.Find the details of the most recent lease.
SELECT * FROM Lease ORDER BY endDate DESC LIMIT 1;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: | Fetch rows:

leaseID	carID	customerID	startDate	endDate	leaseType
10	10	10	2023-10-10	2023-10-31	Monthly
NULL	NULL	NULL	NULL	NULL	NULL

```sql
-- 11.List all payments made in the year 2023
SELECT * FROM Payment WHERE YEAR(paymentDate) = 2023;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

paymentID	leaseID	paymentDate	amount
1	1	2023-01-03	200.00
2	2	2023-02-20	1000.00
NULL	NULL	NULL	NULL

```sql
-- 12.Retrieve customers who have not made any payments.
SELECT * FROM Customer
WHERE customerID NOT IN (SELECT customerID FROM Lease WHERE leaseID IN (SELECT leaseID FROM Payment));
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

customerID	firstName	lastName	email	phoneNumber
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	555-243-5678
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
NULL	NULL	NULL	NULL	NULL

```sql
-- 13.Retrieve Car Details and Their Total Payments
SELECT Vehicle.*, SUM(Payment.amount) AS totalPayments
FROM Vehicle
JOIN Lease ON Vehicle.carID = Lease.carID
JOIN Payment ON Lease.leaseID = Payment.leaseID
GROUP BY Vehicle.carID;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

carID	make	model	year	dailyRate	available	passengerCapacity	engineCapacity	totalPayments
1	Toyota	Camry	2022	50.00	1	4	1450	200.00
2	Honda	Civic	2023	45.00	1	7	1500	1000.00

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

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

customerID	firstName	lastName	email	phoneNumber	totalPayments
1	John	Doe	johndoe@example.com	555-555-5555	200.00
2	Jane	Smith	janesmith@example.com	555-123-4567	1000.00

```sql
-- 15.List Car Details for Each Lease.
SELECT Lease.*, Vehicle.make, Vehicle.model
FROM Lease
JOIN Vehicle ON Lease.carID = Vehicle.carID;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

leaseID	carID	customerID	startDate	endDate	leaseType	make	model
1	1	1	2023-01-01	2023-01-05	Daily	Toyota	Camry
2	2	2	2023-02-15	2023-02-28	Monthly	Honda	Civic
3	3	3	2023-03-10	2023-03-15	Daily	Ford	Focus
4	4	4	2023-04-20	2023-04-30	Monthly	Nissan	Altima
5	5	5	2023-05-05	2023-05-10	Daily	Chevrolet	Malibu
6	4	3	2023-06-15	2023-06-30	Monthly	Nissan	Altima
7	7	7	2023-07-01	2023-07-10	Daily	BMW	3 Series
8	8	8	2023-08-12	2023-08-15	Monthly	Mercedes	C-Class
9	3	3	2023-09-07	2023-09-10	Daily	Ford	Focus
10	10	10	2023-10-10	2023-10-31	Monthly	Lexus	ES

```sql
-- 16.Retrieve Details of Active Leases with Customer and Car Information.
SELECT *
FROM Lease
WHERE Lease.endDate >= CURDATE()
AND carID IN (SELECT carID FROM Vehicle);
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

leaseID	carID	customerID	startDate	endDate	leaseType	make	model
1	1	1	2023-01-01	2023-01-05	Daily	Toyota	Camry
2	2	2	2023-02-15	2023-02-28	Monthly	Honda	Civic
3	3	3	2023-03-10	2023-03-15	Daily	Ford	Focus
4	4	4	2023-04-20	2023-04-30	Monthly	Nissan	Altima
5	5	5	2023-05-05	2023-05-10	Daily	Chevrolet	Malibu
6	4	3	2023-06-15	2023-06-30	Monthly	Nissan	Altima
7	7	7	2023-07-01	2023-07-10	Daily	BMW	3 Series
8	8	8	2023-08-12	2023-08-15	Monthly	Mercedes	C-Class
9	3	3	2023-09-07	2023-09-10	Daily	Ford	Focus
10	10	10	2023-10-10	2023-10-31	Monthly	Lexus	ES

```sql
-- 17.Find the Customer Who Has Spent the Most on Leases.
SELECT Customer.*, SUM(Payment.amount) AS totalSpent
FROM Customer
JOIN Lease ON Customer.customerID = Lease.customerID
JOIN Payment ON Lease.leaseID = Payment.leaseID
GROUP BY Customer.customerID
ORDER BY totalSpent DESC
LIMIT 1;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

leaseID	carID	customerID	startDate	endDate	leaseType
NULL	NULL	NULL	NULL	NULL	NULL

```sql
-- 18.List All Cars with Their Current Lease Information.
SELECT Vehicle.*, Lease.startDate, Lease.endDate, Lease.leaseType
FROM Vehicle
LEFT JOIN Lease ON Vehicle.carID = Lease.carID
WHERE Lease.endDate >= CURDATE() OR Lease.endDate IS NULL;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

carID	make	model	year	dailyRate	available	passengerCapacity	engineCapacity	startDate	endDate	leaseType
6	Hyundai	Sonata	2023	49.00	0	7	1400	NULL	NULL	NULL
9	Audi	A4	2022	55.00	0	4	2500	NULL	NULL	NULL

Date – 04-10-2024