Name: Dheeraj Vemula
Coding Challenge
Paper solved: Coding Challenge-4

Creating Database for car rental system which is carrental.

```
mysql> create database carrental;
Query OK, 1 row affected (0.05 sec)

mysql> use carrental;
Database changed
```

Creating tables named Vehicle, Customer, Lease and Payments.

```
mysql> CREATE TABLE Vehicle (
    ->     vehicleID INT PRIMARY KEY,
    ->     make VARCHAR(100),
    ->     model VARCHAR(100),
    ->     year INT,
    ->     dailyRate DECIMAL(10, 2),
    ->     status BIT,
    ->     passengerCapacity INT,
    ->     engineCapacity INT
    -> );
Query OK, 0 rows affected (0.10 sec)

mysql> CREATE TABLE Customer (
    ->     customerID INT PRIMARY KEY,
    ->     firstName VARCHAR(100),
    ->     lastName VARCHAR(100),
    ->     email VARCHAR(100),
    ->     phoneNumber VARCHAR(20)
    -> );
Query OK, 0 rows affected (0.05 sec)
```

```
mysql> CREATE TABLE Lease (
    ->     leaseID INT PRIMARY KEY,
    ->     vehicleID INT,
    ->     customerID INT,
    ->     startDate DATE,
    ->     endDate DATE,
    ->     type VARCHAR(20) CHECK (type IN ('DailyLease', 'MonthlyLease')),
    ->     FOREIGN KEY (vehicleID) REFERENCES Vehicle(vehicleID),
    ->     FOREIGN KEY (customerID) REFERENCES Customer(customerID)
    -> );
Query OK, 0 rows affected (0.10 sec)
```

```
mysql> CREATE TABLE Payment (
    ->     paymentID INT PRIMARY KEY,
    ->     leaseID INT,
    ->     paymentDate DATE,
    ->     amount DECIMAL(10, 2),
    ->     FOREIGN KEY (leaseID) REFERENCES Lease(leaseID)
    -> );
Query OK, 0 rows affected (0.06 sec)
```

Inserting values into vehicle, customer, lease and payment tables.

```
mysql> 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);
Query OK, 10 rows affected (0.02 sec)
Records: 10  Duplicates: 0  Warnings: 0

mysql> 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');
Query OK, 10 rows affected (0.01 sec)
Records: 10  Duplicates: 0  Warnings: 0
mysql> 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');
Query OK, 10 rows affected (0.01 sec)
Records: 10  Duplicates: 0  Warnings: 0
```

```
mysql> 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);
Query OK, 10 rows affected (0.01 sec)
Records: 10  Duplicates: 0  Warnings: 0
```

Tasks:

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

```
mysql> UPDATE Vehicle
    -> SET dailyRate = 68.00
    -> WHERE make = 'Mercedes';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

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

```
mysql> DELETE FROM Payment
    -> WHERE leaseID IN (SELECT leaseID FROM Lease WHERE customerID = 4);
Query OK, 1 row affected (0.01 sec)

mysql>
mysql> DELETE FROM Lease
    -> WHERE customerID = 4;
Query OK, 1 row affected (0.01 sec)

mysql>
mysql> DELETE FROM Customer
    -> WHERE customerID = 4;
Query OK, 1 row affected (0.01 sec)
```

```
mysql> select * from Payment;
+-----------+---------+-----------------+---------+
| paymentID | leaseID | transactionDate | amount  |
+-----------+---------+-----------------+---------+
|         1 |       1 | 2023-01-03      |  200.00 |
|         2 |       2 | 2023-02-20      | 1000.00 |
|         3 |       3 | 2023-03-12      |   75.00 |
|         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 |
+-----------+---------+-----------------+---------+
9 rows in set (0.00 sec)
```

Payments and leases related to customerID =4 is deleted.

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

```
mysql> ALTER TABLE Payment
    -> RENAME COLUMN paymentDate TO transactionDate;
Query OK, 0 rows affected (0.04 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

4. Find a specific customer by email.

```
mysql> SELECT * FROM Customer
    -> WHERE email ='robert@example.com';
+------------+-----------+----------+--------------------+--------------+
| customerID | firstName | lastName | email              | phoneNumber  |
+------------+-----------+----------+--------------------+--------------+
|          3 | Robert    | Johnson  | robert@example.com | 555-789-1234 |
+------------+-----------+----------+--------------------+--------------+
1 row in set (0.00 sec)
```

5. Get active leases for a specific customer.

```
mysql> SELECT * FROM Lease
    -> WHERE customerID =1
    -> AND endDate <= CURRENT_DATE;
+---------+-----------+------------+------------+------------+------------+
| leaseID | vehicleID | customerID | startDate  | endDate    | type       |
+---------+-----------+------------+------------+------------+------------+
|       1 |         1 |          1 | 2023-01-01 | 2023-01-05 | DailyLease |
+---------+-----------+------------+------------+------------+------------+
1 row in set (0.00 sec)
```

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

```
mysql> SELECT C.firstName, C.lastName, P.* FROM Payment P
    -> JOIN Lease L ON P.leaseID = L.leaseID
    -> JOIN Customer C ON L.customerID = C.customerID
    -> WHERE C.phoneNumber ='555-789-1234';
+-----------+----------+-----------+---------+-----------------+---------+
| firstName | lastName | paymentID | leaseID | transactionDate | amount  |
+-----------+----------+-----------+---------+-----------------+---------+
| Robert    | Johnson  |         3 |       3 | 2023-03-12      |   75.00 |
| Robert    | Johnson  |         6 |       6 | 2023-06-18      | 1200.00 |
| Robert    | Johnson  |         9 |       9 | 2023-09-09      |   80.00 |
+-----------+----------+-----------+---------+-----------------+---------+
3 rows in set (0.00 sec)
```

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

```
mysql> SELECT AVG(dailyRate) AS averageDailyRate
    -> FROM Vehicle WHERE status = 'available';
+------------------+
| averageDailyRate |
+------------------+
|        50.666667 |
+------------------+
1 row in set, 1 warning (0.00 sec)
```

8. Find the car with the highest daily rate.

```
mysql> SELECT *FROM Vehicle
    -> ORDER BY dailyRate DESC LIMIT 1;
+-----------+----------+---------+------+-----------+---------------+------------------+----------------+
| vehicleID | make     | model   | year | dailyRate | status        | passengerCapacity | engineCapacity |
+-----------+----------+---------+------+-----------+---------------+------------------+----------------+
|         8 | Mercedes | C-Class | 2022 |     68.00 | 0x01          |                8 |           2599 |
+-----------+----------+---------+------+-----------+---------------+------------------+----------------+
1 row in set (0.00 sec)
```

9. Retrieve all cars leased by a specific customer:

```
mysql> SELECT V.* FROM Vehicle V
    -> JOIN Lease L ON V.vehicleID = L.vehicleID
    -> WHERE L.customerID = 2;
+-----------+-------+-------+------+-----------+---------------+------------------+----------------+
| vehicleID | make  | model | year | dailyRate | status        | passengerCapacity | engineCapacity |
+-----------+-------+-------+------+-----------+---------------+------------------+----------------+
|         2 | Honda | Civic | 2023 |     45.00 | 0x01          |                7 |           1500 |
+-----------+-------+-------+------+-----------+---------------+------------------+----------------+
1 row in set (0.00 sec)
```

10. Find the details of the most recent lease:

```
mysql> SELECT V.make, V.model, L.*
    -> FROM Vehicle V
    -> JOIN Lease L ON V.vehicleID = L.vehicleID
    -> ORDER BY L.endDate DESC
    -> LIMIT 1;
+-------+-------+---------+-----------+------------+------------+------------+--------------+
| make  | model | leaseID | vehicleID | customerID | startDate  | endDate    | type         |
+-------+-------+---------+-----------+------------+------------+------------+--------------+
| Lexus | ES    |      10 |        10 |         10 | 2023-10-10 | 2023-10-31 | MonthlyLease |
+-------+-------+---------+-----------+------------+------------+------------+--------------+
1 row in set (0.00 sec)
```

11. List all payments made in the year 2023:

```
mysql> SELECT * FROM Payment
    -> WHERE YEAR(transactionDate) = 2023;
+-----------+---------+-----------------+---------+
| paymentID | leaseID | transactionDate | amount  |
+-----------+---------+-----------------+---------+
|         1 |       1 | 2023-01-03      |  200.00 |
|         2 |       2 | 2023-02-20      | 1000.00 |
|         3 |       3 | 2023-03-12      |   75.00 |
|         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 |
+-----------+---------+-----------------+---------+
9 rows in set (0.00 sec)
```

12. Retrieve customers who have not made any payments:

```
mysql> SELECT * FROM Customer
    -> WHERE customerID NOT IN (SELECT DISTINCT customerID FROM Lease);
+------------+-----------+----------+---------------------+--------------+
| customerID | firstName | lastName | email               | phoneNumber  |
+------------+-----------+----------+---------------------+--------------+
|          6 | Laura     | Hall     | laura@example.com   | 555-234-5678 |
|          9 | William   | Taylor   | william@example.com | 555-321-6547 |
+------------+-----------+----------+---------------------+--------------+
2 rows in set (0.00 sec)
```

13. Retrieve Car Details and Their Total Payments:

```
mysql> SELECT V.*, COALESCE(SUM(P.amount),0) AS totalPayments
    -> FROM Vehicle V
    -> LEFT JOIN Lease L ON V.vehicleID = L.vehicleID
    -> LEFT JOIN Payment P ON L.leaseID = P.leaseID
    -> GROUP BY V.vehicleID;
+-----------+----------+----------+------+-----------+--------------+-------------------+----------------+--------------+
| vehicleID | make     | model    | year | dailyRate | status       | passengerCapacity | engineCapacity | totalPayments |
+-----------+----------+----------+------+-----------+--------------+-------------------+----------------+--------------+
|         1 | Toyota   | Camry    | 2022 |     50.00 | 0x01         |                 4 |           1450 |       200.00 |
|         2 | Honda    | Civic    | 2023 |     45.00 | 0x01         |                 7 |           1500 |      1000.00 |
|         3 | Ford     | Focus    | 2022 |     48.00 | 0x00         |                 4 |           1400 |       155.00 |
|         4 | Nissan   | Altima   | 2023 |     52.00 | 0x01         |                 7 |           1200 |      1200.00 |
|         5 | Chevrolet| Malibu   | 2022 |     47.00 | 0x01         |                 4 |           1800 |        60.00 |
|         6 | Hyundai  | Sonata   | 2023 |     49.00 | 0x00         |                 7 |           1400 |         0.00 |
|         7 | BMW      | 3 Series | 2023 |     60.00 | 0x01         |                 7 |           2499 |        40.00 |
|         8 | Mercedes | C-Class  | 2022 |     68.00 | 0x01         |                 8 |           2599 |      1100.00 |
|         9 | Audi     | A4       | 2022 |     55.00 | 0x00         |                 4 |           2500 |         0.00 |
|        10 | Lexus    | ES       | 2023 |     54.00 | 0x01         |                 4 |           2500 |      1500.00 |
+-----------+----------+----------+------+-----------+--------------+-------------------+----------------+--------------+
10 rows in set (0.00 sec)
```

14. Calculate Total Payments for Each Customer:

```
mysql> SELECT C.customerID, C.firstName, C.lastName, COALESCE(SUM(P.amount),0) AS totalPayments
    -> FROM Customer C
    -> LEFT JOIN Lease L ON C.customerID = L.customerID
    -> LEFT JOIN Payment P ON L.leaseID = P.leaseID
    -> GROUP BY C.customerID, C.firstName, C.lastName;
+------------+-----------+----------+---------------+
| customerID | firstName | lastName | totalPayments |
+------------+-----------+----------+---------------+
|          1 | John      | Doe      |        200.00 |
|          2 | Jane      | Smith    |       1000.00 |
|          3 | Robert    | Johnson  |       1355.00 |
|          5 | David     | Lee      |         60.00 |
|          6 | Laura     | Hall     |          0.00 |
|          7 | Michael   | Davis    |         40.00 |
|          8 | Emma      | Wilson   |       1100.00 |
|          9 | William   | Taylor   |          0.00 |
|         10 | Olivia    | Adams    |       1500.00 |
+------------+-----------+----------+---------------+
9 rows in set (0.00 sec)
```

15. List Car Details for Each Lease

```
mysql> SELECT L.*, V.make, V.model
    -> FROM Lease L
    -> JOIN Vehicle V ON L.vehicleID = V.vehicleID;
+---------+-----------+------------+------------+------------+-------------+-----------+----------+
| leaseID | vehicleID | customerID | startDate  | endDate    | type        | make      | model    |
+---------+-----------+------------+------------+------------+-------------+-----------+----------+
|       1 |         1 |          1 | 2023-01-01 | 2023-01-05 | DailyLease  | Toyota    | Camry    |
|       2 |         2 |          2 | 2023-02-15 | 2023-02-28 | MonthlyLease| Honda     | Civic    |
|       3 |         3 |          3 | 2023-03-10 | 2023-03-15 | DailyLease  | Ford      | Focus    |
|       5 |         5 |          5 | 2023-05-05 | 2023-05-10 | DailyLease  | Chevrolet | Malibu   |
|       6 |         4 |          3 | 2023-06-15 | 2023-06-30 | MonthlyLease| Nissan    | Altima   |
|       7 |         7 |          7 | 2023-07-01 | 2023-07-10 | DailyLease  | BMW       | 3 Series |
|       8 |         8 |          8 | 2023-08-12 | 2023-08-15 | MonthlyLease| Mercedes  | C-Class  |
|       9 |         3 |          3 | 2023-09-07 | 2023-09-10 | DailyLease  | Ford      | Focus    |
|      10 |        10 |         10 | 2023-10-10 | 2023-10-31 | MonthlyLease| Lexus     | ES       |
+---------+-----------+------------+------------+------------+-------------+-----------+----------+
```

16. Retrieve Details of Active Leases with Customer and Car Information:

```
mysql> SELECT L.*, C.*, V.*
    -> FROM Lease L
    -> JOIN Customer C ON L.customerID = C.customerID
    -> JOIN Vehicle V ON L.vehicleID = V.vehicleID
    -> WHERE L.endDate >= CURDATE();
Empty set (0.00 sec)
```

17. Find the Customer Who Has Spent the Most on Leases:

```
mysql> SELECT C.*, SUM(P.amount) AS totalSpentOnLeases
    -> 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 totalSpentOnLeases DESC
    -> LIMIT 1;
+------------+-----------+----------+--------------------+--------------+---------------------+
| customerID | firstName | lastName | email              | phoneNumber  | totalSpentOnLeases  |
+------------+-----------+----------+--------------------+--------------+---------------------+
|         10 | Olivia    | Adams    | olivia@example.com | 555-765-4321 |             1500.00 |
+------------+-----------+----------+--------------------+--------------+---------------------+
1 row in set (0.00 sec)
```

18. List All Cars with Their Current Lease Information:

```
mysql> SELECT V.*, L.* FROM Vehicle V
    -> LEFT JOIN Lease L ON V.vehicleID = L.vehicleID AND L.endDate >= CURDATE();
+-----------+----------+----------+------+-----------+---------------+------------------+---------------+---------+------
-----+------------+-----------+---------+------+
| vehicleID | make     | model    | year | dailyRate | status        | passengerCapacity | engineCapacity | leaseID | vehic
leID | customerID | startDate | endDate | type |
+-----------+----------+----------+------+-----------+---------------+------------------+---------------+---------+------
-----+------------+-----------+---------+------+
|         1 | Toyota   | Camry    | 2022 |     50.00 | 0x01          |                4 |          1450 |    NULL |
NULL |       NULL | NULL      |    NULL | NULL |
|         2 | Honda    | Civic    | 2023 |     45.00 | 0x01          |                7 |          1500 |    NULL |
NULL |       NULL | NULL      |    NULL | NULL |
|         3 | Ford     | Focus    | 2022 |     48.00 | 0x00          |                4 |          1400 |    NULL |
NULL |       NULL | NULL      |    NULL | NULL |
|         4 | Nissan   | Altima   | 2023 |     52.00 | 0x01          |                7 |          1200 |    NULL |
NULL |       NULL | NULL      |    NULL | NULL |
|         5 | Chevrolet | Malibu  | 2022 |     47.00 | 0x01          |                4 |          1800 |    NULL |
NULL |       NULL | NULL      |    NULL | NULL |
|         6 | Hyundai  | Sonata   | 2023 |     49.00 | 0x00          |                7 |          1400 |    NULL |
NULL |       NULL | NULL      |    NULL | NULL |
|         7 | BMW      | 3 Series | 2023 |     60.00 | 0x01          |                7 |          2499 |    NULL |
NULL |       NULL | NULL      |    NULL | NULL |
|         8 | Mercedes | C-Class  | 2022 |     68.00 | 0x01          |                8 |          2599 |    NULL |
NULL |       NULL | NULL      |    NULL | NULL |
|         9 | Audi     | A4       | 2022 |     55.00 | 0x00          |                4 |          2500 |    NULL |
NULL |       NULL | NULL      |    NULL | NULL |
|        10 | Lexus    | ES       | 2023 |     54.00 | 0x01          |                4 |          2500 |    NULL |
NULL |       NULL | NULL      |    NULL | NULL |
+-----------+----------+----------+------+-----------+---------------+------------------+---------------+---------+------
-----+------------+-----------+---------+------+
10 rows in set (0.00 sec)
```