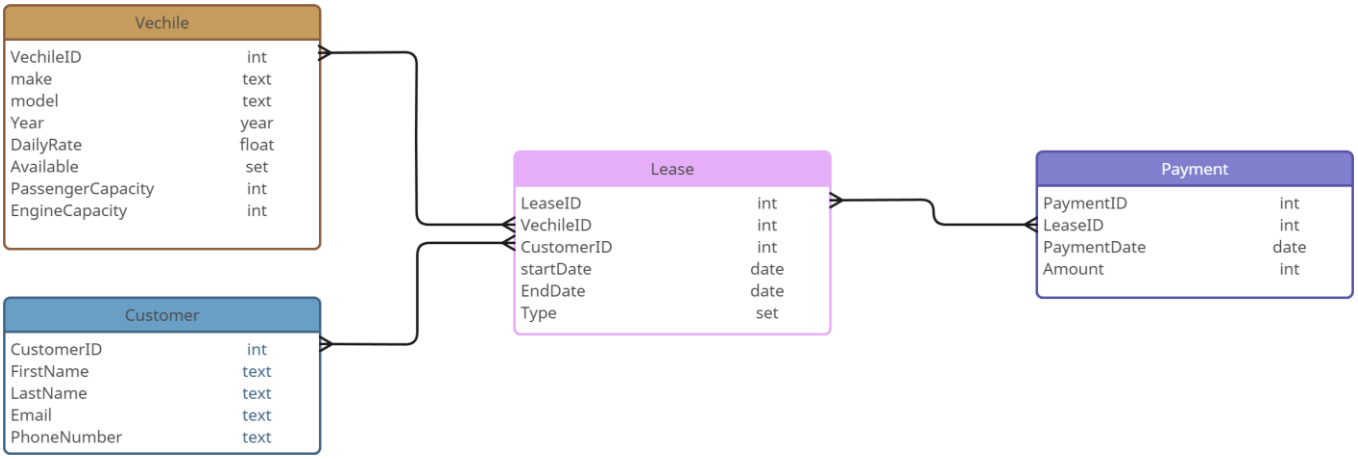
Coding Challenge - Car Rental System – SQL

Creating Database

```
mysql> create database CarRentalSystem
    -> ;
Query OK, 1 row affected (0.02 sec)
```

## ER DIAGRAM




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

```
mysql> UPDATE Vechile SET Dailyrate=68 WHERE make = 'mercedes';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> SELECT * FROM Vechile;
+-----------+----------+----------+------+-----------+-----------+-------------------+----------------+
| VechileID | make     | model    | year | dailyRate | available | PassengerCapacity | EngineCapacity |
+-----------+----------+----------+------+-----------+-----------+-------------------+----------------+
|         1 | Toyota   | Camry    | 2022 |        50 | 1         |                 4 |           1450 |
|         2 | Honda    | Civic    | 2023 |        45 | 1         |                 7 |           1500 |
|         3 | Ford     | Focus    | 2022 |        48 | 0         |                 4 |           1400 |
|         4 | Nissan   | Altima   | 2023 |        52 | 1         |                 7 |           1200 |
|         5 | Chevrolet| Malibu   | 2022 |        47 | 1         |                 4 |           1800 |
|         6 | Hyundai  | Sonata   | 2023 |        49 | 0         |                 7 |           1400 |
|         7 | BMW      | 3 Series | 2023 |        60 | 1         |                 7 |           2499 |
|         8 | Mercedes | C-Class  | 2022 |        68 | 1         |                 8 |           2599 |
|         9 | Audi     | A4       | 2022 |        55 | 0         |                 4 |           2500 |
|        10 | Lexus    | ES       | 2023 |        54 | 1         |                 4 |           2500 |
+-----------+----------+----------+------+-----------+-----------+-------------------+----------------+
10 rows in set (0.00 sec)
```

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

**I HAVE USED ON DELETE CASCADE ON UPDATE CASCADE TO MAINTAIN REFERENCIAL INTEGRITY**

```
mysql> DELETE FROM Customer WHERE CustomerID=3;
Query OK, 1 row affected (0.01 sec)

mysql> SELECT * FROM Payment;
+-----------+---------+-------------+--------+
| PaymentID | LeaseID | PaymentDate | amount |
+-----------+---------+-------------+--------+
|         1 |       1 | 2023-01-03  |    200 |
|         2 |       2 | 2023-02-20  |   1000 |
|         4 |       4 | 2023-04-25  |    900 |
|         5 |       5 | 2023-05-07  |     60 |
|         7 |       7 | 2023-07-03  |     40 |
|         8 |       8 | 2023-08-14  |   1100 |
|        10 |      10 | 2023-10-25  |   1500 |
+-----------+---------+-------------+--------+
7 rows in set (0.00 sec)

mysql> SELECT * FROM Lease;
+---------+-----------+------------+------------+------------+---------+
| LeaseID | VechileID | CustomerID | startDate  | endDate    | type    |
+---------+-----------+------------+------------+------------+---------+
|       1 |         1 |          1 | 2023-01-01 | 2023-01-05 | Daily   |
|       2 |         2 |          2 | 2023-02-15 | 2023-02-28 | Monthly |
|       4 |         4 |          4 | 2023-04-20 | 2023-04-30 | Monthly |
|       5 |         5 |          5 | 2023-05-05 | 2023-05-10 | Daily   |
|       7 |         7 |          7 | 2023-07-01 | 2023-07-10 | Daily   |
|       8 |         8 |          8 | 2023-08-12 | 2023-08-15 | Monthly |
|      10 |        10 |         10 | 2023-10-10 | 2023-10-31 | Monthly |
+---------+-----------+------------+------------+------------+---------+
7 rows in set (0.00 sec)
```

```
mysql> SELECT * FROM Customer;
+------------+-----------+----------+-----------------------+--------------+
| CustomerID | FirstName | LastName | Email                 | phoneNumber  |
+------------+-----------+----------+-----------------------+--------------+
|          1 | John      | Doe      | johndoe@example.com   | 555-555-5555 |
|          2 | Jane      | Smith    | janesmith@example.com | 555-123-4567 |
|          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 |
+------------+-----------+----------+-----------------------+--------------+
9 rows in set (0.00 sec)
```

**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 (0.02 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM Payment;
+-----------+---------+-----------------+--------+
| PaymentID | LeaseID | TransactionDate | amount |
+-----------+---------+-----------------+--------+
|         1 |       1 | 2023-01-03      |    200 |
|         2 |       2 | 2023-02-20      |   1000 |
|         4 |       4 | 2023-04-25      |    900 |
|         5 |       5 | 2023-05-07      |     60 |
|         7 |       7 | 2023-07-03      |     40 |
|         8 |       8 | 2023-08-14      |   1100 |
|        10 |      10 | 2023-10-25      |   1500 |
+-----------+---------+-----------------+--------+
7 rows in set (0.00 sec)
```

**4. Find a specific customer by email.**

```
mysql> DELIMITER ;
mysql> CALL callCustomer('laura@example.com');
+------------+-----------+----------+--------------------+--------------+
| CustomerID | FirstName | LastName | Email              | phoneNumber  |
+------------+-----------+----------+--------------------+--------------+
|          6 | Laura     | Hall     | laura@example.com | 555-234-5678 |
+------------+-----------+----------+--------------------+--------------+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.01 sec)
```

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

```
mysql> SELECT C.customerID,CONCAT(C.FirstName,' ',C.LastName),L.* FROM
    -> Customer C JOIN Lease L ON
    -> C.customerID=L.customerID
    -> GROUP BY C.customerID,L.LeaseID;
+------------+-------------------------------------+---------+-----------+------------+------------+------------+---------+
| customerID | CONCAT(C.FirstName,' ',C.LastName) | LeaseID | VechileID | CustomerID | startDate  | endDate    | type    |
+------------+-------------------------------------+---------+-----------+------------+------------+------------+---------+
|          1 | John Doe                            |       1 |         1 |          1 | 2023-01-01 | 2023-01-05 | Daily   |
|          2 | Jane Smith                          |       2 |         2 |          2 | 2023-02-15 | 2023-02-28 | Monthly |
|          4 | Sarah Brown                         |       4 |         4 |          4 | 2023-04-20 | 2023-04-30 | Monthly |
|          5 | David Lee                           |       5 |         5 |          5 | 2023-05-05 | 2023-05-10 | Daily   |
|          7 | Michael Davis                       |       7 |         7 |          7 | 2023-07-01 | 2023-07-10 | Daily   |
|          8 | Emma Wilson                         |       8 |         8 |          8 | 2023-08-12 | 2023-08-15 | Monthly |
|         10 | Olivia Adams                        |      10 |        10 |         10 | 2023-10-10 | 2023-10-31 | Monthly |
+------------+-------------------------------------+---------+-----------+------------+------------+------------+---------+
7 rows in set (0.00 sec)
```

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

```
mysql> INSERT INTO Payment VALUES(11,1,'2024-01-15',500);
Query OK, 1 row affected (0.01 sec)

mysql> SELECT P.* FROM
    -> Payment P JOIN Lease L
    -> ON P.LeaseID=L.LeaseID
    -> WHERE L.CustomerID=(SELECT customerID FROM Customer WHERE PhoneNumber ='555-555-5555');
+-----------+---------+-----------------+--------+
| PaymentID | LeaseID | TransactionDate | amount |
+-----------+---------+-----------------+--------+
|         1 |       1 | 2023-01-03      |    200 |
|        11 |       1 | 2024-01-15      |    500 |
+-----------+---------+-----------------+--------+
2 rows in set (0.00 sec)
```

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

```
mysql> SELECT AVG(DailyRate) AS 'AVERAGE DAILY RATE FOR AVAILABLE CARS'
    -> FROM Vechile WHERE available='1';
+----------------------------------------+
| AVERAGE DAILY RATE FOR AVAILABLE CARS |
+----------------------------------------+
|                    53.714285714285715 |
+----------------------------------------+
1 row in set (0.00 sec)
```

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

```
mysql> SELECT * FROM VECHILE
    -> WHERE DailyRate=(SELECT MAX(DailyRate) FROM Vechile);
+-----------+----------+---------+------+-----------+-----------+-------------------+----------------+
| VechileID | make     | model   | year | dailyRate | available | PassengerCapacity | EngineCapacity |
+-----------+----------+---------+------+-----------+-----------+-------------------+----------------+
|         8 | Mercedes | C-Class | 2022 |        68 | 1         |                 8 |           2599 |
+-----------+----------+---------+------+-----------+-----------+-------------------+----------------+
1 row in set (0.00 sec)
```

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

```
mysql> DELIMITER ##
mysql> CREATE PROCEDURE CustomerID(IN Num int)
    -> BEGIN
    -> SELECT V.* FROM Vechile V JOIN Lease L
    -> ON V.vechileID=L.vechileID
    -> WHERE L.customerID=Num;
    -> END ##
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> DELIMITER ;
mysql> CALL CustomerID(2);
+-----------+-------+-------+------+-----------+-----------+-------------------+----------------+
| VechileID | make  | model | year | dailyRate | available | PassengerCapacity | EngineCapacity |
+-----------+-------+-------+------+-----------+-----------+-------------------+----------------+
|         2 | Honda | Civic | 2023 |        45 | 1         |                 7 |           1500 |
+-----------+-------+-------+------+-----------+-----------+-------------------+----------------+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.01 sec)
```

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

```
mysql> SELECT * FROM Lease
    -> ORDER BY startDate DESC LIMIT 1;
+---------+-----------+------------+------------+------------+---------+
| LeaseID | VechileID | CustomerID | startDate  | endDate    | type    |
+---------+-----------+------------+------------+------------+---------+
|      10 |        10 |         10 | 2023-10-10 | 2023-10-31 | Monthly |
+---------+-----------+------------+------------+------------+---------+
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 |
|         2 |       2 | 2023-02-20      |   1000 |
|         4 |       4 | 2023-04-25      |    900 |
|         5 |       5 | 2023-05-07      |     60 |
|         7 |       7 | 2023-07-03      |     40 |
|         8 |       8 | 2023-08-14      |   1100 |
|        10 |      10 | 2023-10-25      |   1500 |
+-----------+---------+-----------------+--------+
7 rows in set (0.00 sec)
```

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

```
mysql> SELECT * FROM CUSTOMER WHERE
    -> CustomerID NOT IN (SELECT L.CustomerID
    -> FROM Lease L JOIN Payment P ON
    -> L.leaseID=P.LeaseID);
+------------+-----------+----------+----------------------+--------------+
| 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.**

**TO DO THIS QUESTION WE HAVE TWO WAYS**

**1.**

```
mysql> SELECT V.*,SUM(A.amount) AS 'Total Payment' FROM
    -> Vechile V JOIN (SELECT L.vechileID,P.amount FROM Lease L JOIN
    -> Payment P ON L.leaseID=P.LeaseID ) A ON
    -> V.vechileID=A.vechileID
    -> GROUP BY VechileID;
+-----------+-----------+----------+------+-----------+-----------+-------------------+----------------+---------------+
| VechileID | make      | model    | year | dailyRate | available | PassengerCapacity | EngineCapacity | Total Payment |
+-----------+-----------+----------+------+-----------+-----------+-------------------+----------------+---------------+
|         1 | Toyota    | Camry    | 2022 |        50 | 1         |                 4 |           1450 |           700 |
|         2 | Honda     | Civic    | 2023 |        45 | 1         |                 7 |           1500 |          1000 |
|         4 | Nissan    | Altima   | 2023 |        52 | 1         |                 7 |           1200 |           900 |
|         5 | Chevrolet | Malibu   | 2022 |        47 | 1         |                 4 |           1800 |            60 |
|         7 | BMW       | 3 Series | 2023 |        60 | 1         |                 7 |           2499 |            40 |
|         8 | Mercedes  | C-Class  | 2022 |        68 | 1         |                 8 |           2599 |          1100 |
|        10 | Lexus     | ES       | 2023 |        54 | 1         |                 4 |           2500 |          1500 |
+-----------+-----------+----------+------+-----------+-----------+-------------------+----------------+---------------+
7 rows in set (0.00 sec)
```

**2.**

```
mysql> SELECT V.*, SUM(datediff(L.endDate,L.startDate)*V.dailyrate) AS 'Total Payment had to be Done'
    -> FROM Vechile V JOIN Lease L ON V.VechileID=L.VechileID
    -> GROUP BY V.vechileID;
+-----------+-----------+----------+------+-----------+-----------+-------------------+----------------+-----------------------------+
| VechileID | make      | model    | year | dailyRate | available | PassengerCapacity | EngineCapacity | Total Payment had to be Done |
+-----------+-----------+----------+------+-----------+-----------+-------------------+----------------+-----------------------------+
|         1 | Toyota    | Camry    | 2022 |        50 | 1         |                 4 |           1450 |                         200 |
|         2 | Honda     | Civic    | 2023 |        45 | 1         |                 7 |           1500 |                         585 |
|         4 | Nissan    | Altima   | 2023 |        52 | 1         |                 7 |           1200 |                         520 |
|         5 | Chevrolet | Malibu   | 2022 |        47 | 1         |                 4 |           1800 |                         235 |
|         7 | BMW       | 3 Series | 2023 |        60 | 1         |                 7 |           2499 |                         540 |
|         8 | Mercedes  | C-Class  | 2022 |        68 | 1         |                 8 |           2599 |                         204 |
|        10 | Lexus     | ES       | 2023 |        54 | 1         |                 4 |           2500 |                        1134 |
+-----------+-----------+----------+------+-----------+-----------+-------------------+----------------+-----------------------------+
7 rows in set (0.00 sec)
```

**14. Calculate Total Payments for Each Customer.**

```
mysql> SELECT C.*,SUM(A.amount) AS 'Total Payment' FROM
    -> Customer C JOIN (SELECT L.customerID,P.amount FROM Lease L JOIN
    -> Payment P ON L.leaseID=P.LeaseID ) A ON
    -> C.CustomerID=A.CustomerID
    -> GROUP BY CustomerID;
+------------+-----------+----------+-----------------------+--------------+----------------+
| CustomerID | FirstName | LastName | Email                 | phoneNumber  | Total Payment  |
+------------+-----------+----------+-----------------------+--------------+----------------+
|          1 | John      | Doe      | johndoe@example.com   | 555-555-5555 |            700 |
|          2 | Jane      | Smith    | janesmith@example.com | 555-123-4567 |           1000 |
|          4 | Sarah     | Brown    | sarah@example.com     | 555-456-7890 |            900 |
|          5 | David     | Lee      | david@example.com     | 555-987-6543 |             60 |
|          7 | Michael   | Davis    | michael@example.com   | 555-876-5432 |             40 |
|          8 | Emma      | Wilson   | emma@example.com      | 555-432-1098 |           1100 |
|         10 | Olivia    | Adams    | olivia@example.com    | 555-765-4321 |           1500 |
+------------+-----------+----------+-----------------------+--------------+----------------+
7 rows in set (0.00 sec)
```

**15. List Car Details for Each Lease.**

```
mysql> SELECT L.*,V.* FROM Lease L
    -> JOIN Vechile V ON L.vechileId=V.vechileID
    -> GROUP BY L.leaseID,V.VechileID;
+---------+----------+------------+-----------+-----------+---------+-----------+---------+---------+------+-----------+-----------+--------------------+----------------+
| LeaseID | VechileID | CustomerID | startDate | endDate   | type    | VechileID | make    | model   | year | dailyRate | available | PassengerCapacity | EngineCapacity |
+---------+----------+------------+-----------+-----------+---------+-----------+---------+---------+------+-----------+-----------+--------------------+----------------+
|       1 |        1 |          1 | 2023-01-01 | 2023-01-05 | Daily   |         1 | Toyota  | Camry   | 2022 |        50 | 1         |                  4 |           1450 |
|       2 |        2 |          2 | 2023-02-15 | 2023-02-28 | Monthly |         2 | Honda   | Civic   | 2023 |        45 | 1         |                  7 |           1500 |
|       4 |        4 |          4 | 2023-04-20 | 2023-04-30 | Monthly |         4 | Nissan  | Altima  | 2023 |        52 | 1         |                  7 |           1200 |
|       5 |        5 |          5 | 2023-05-05 | 2023-05-10 | Daily   |         5 | Chevrolet | Malibu | 2022 |        47 | 1         |                  4 |           1800 |
|       7 |        7 |          7 | 2023-07-01 | 2023-07-10 | Daily   |         7 | BMW     | 3 Series | 2023 |        60 | 1         |                  7 |           2499 |
|       8 |        8 |          8 | 2023-08-12 | 2023-08-15 | Monthly |         8 | Mercedes | C-Class | 2022 |        68 | 1         |                  8 |           2599 |
|      10 |       10 |         10 | 2023-10-10 | 2023-10-31 | Monthly |        10 | Lexus   | ES      | 2023 |        54 | 1         |                  4 |           2500 |
+---------+----------+------------+-----------+-----------+---------+-----------+---------+---------+------+-----------+-----------+--------------------+----------------+
7 rows in set (0.00 sec)
```

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

```
mysql> SELECT C.*,V.* FROM Customer C,Vechile V
    -> WHERE (v.vechileID,C.customerID) IN
    -> (SELECT VechileID,CustomerID FROM Lease WHERE EndDate >=CURDATE());
+------------+-----------+----------+-----------------+--------------+-----------+------+-------+------+-----------+-----------+--------------------+----------------+
| CustomerID | FirstName | LastName | Email           | phoneNumber  | VechileID | make | model | year | dailyRate | available | PassengerCapacity | EngineCapacity |
+------------+-----------+----------+-----------------+--------------+-----------+------+-------+------+-----------+-----------+--------------------+----------------+
|          4 | Sarah     | Brown    | sarah@example.com | 555-456-7890 |         2 | Honda | Civic | 2023 |        45 | 1         |                  7 |           1500 |
+------------+-----------+----------+-----------------+--------------+-----------+------+-------+------+-----------+-----------+--------------------+----------------+
1 row in set (0.00 sec)
```

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

```
mysql> SELECT C.* , A.TotalPayment FROM
    -> Customer C JOIN (SELECT L.customerID,
    -> SUM(IF(type = 'daily', DATEDIFF(L.endDate, L.startDate) * V.dailyrate, 30 * V.dailyrate)) AS TotalPayment
    -> FROM Vechile V
    -> JOIN Lease L ON V.VechileID = L.VechileID
    -> GROUP BY L.customerID
    -> ) A ON C.customerId=A.customerID
    -> ORDER BY A.totalPayment DESC LIMIT 1;
+------------+-----------+----------+-------------------+--------------+--------------+
| CustomerID | FirstName | LastName | Email             | phoneNumber  | TotalPayment |
+------------+-----------+----------+-------------------+--------------+--------------+
|          4 | Sarah     | Brown    | sarah@example.com | 555-456-7890 |         2235 |
+------------+-----------+----------+-------------------+--------------+--------------+
1 row in set (0.00 sec)
```

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

```
mysql> SELECT V.VechileID,V.Make,V.Model,L.* FROM
    -> Vechile V JOIN Lease L
    -> ON V.vechileID=L.vechileID
    -> ORDER BY V.vechileId;
+-----------+-----------+----------+---------+-----------+------------+------------+------------+---------+
| VechileID | Make      | Model    | LeaseID | VechileID | CustomerID | startDate  | endDate    | type    |
+-----------+-----------+----------+---------+-----------+------------+------------+------------+---------+
|         1 | Toyota    | Camry    |       1 |         1 |          1 | 2023-01-01 | 2023-01-05 | Daily   |
|         2 | Honda     | Civic    |       2 |         2 |          2 | 2023-02-15 | 2023-02-28 | Monthly |
|         2 | Honda     | Civic    |      11 |         2 |          4 | 2024-01-10 | 2024-01-25 | Daily   |
|         4 | Nissan    | Altima   |       4 |         4 |          4 | 2023-04-20 | 2023-04-30 | Monthly |
|         5 | Chevrolet | Malibu   |       5 |         5 |          5 | 2023-05-05 | 2023-05-10 | Daily   |
|         7 | BMW       | 3 Series |       7 |         7 |          7 | 2023-07-01 | 2023-07-10 | Daily   |
|         8 | Mercedes  | C-Class  |       8 |         8 |          8 | 2023-08-12 | 2023-08-15 | Monthly |
|        10 | Lexus     | ES       |      10 |        10 |         10 | 2023-10-10 | 2023-10-31 | Monthly |
+-----------+-----------+----------+---------+-----------+------------+------------+------------+---------+
8 rows in set (0.00 sec)
```