

# CarRental

## Q1. Update the daily rate for a Mercedes car to 68.

The screenshot shows the MySQL Workbench interface with a query editor window titled "Query 1". The code entered is:

```
673 • INSERT INTO Payments (paymentID, leaseID, paymentDate, amount) VALUES
674     (1, 1, '2023-01-03', 200.00),
675     (2, 2, '2023-02-20', 1000.00),
676     (3, 3, '2023-03-12', 75.00),
677     (4, 4, '2023-04-25', 900.00),
678     (5, 5, '2023-05-07', 60.00),
679     (6, 6, '2023-06-18', 1200.00),
680     (7, 7, '2023-07-03', 40.00),
681     (8, 8, '2023-08-14', 1100.00),
682     (9, 9, '2023-09-09', 80.00),
683     (10, 10, '2023-10-25', 1500.00);

684
685
686 • UPDATE Vechicles
687     SET dailyRate = 68.00
688     WHERE make = 'Mercedes';
689
```

The output pane shows the results of the update query:

Action	Time	Message	Duration / Fetch
UPDATE Vechicles SET dailyRate = 68.00 WHERE make = 'Mercedes'	1 21:43:42	0 row(s) affected Rows matched: 1 Changed: 0 Warnings: 0	0.000 sec

## Q2. Delete a specific customer and all associated leases and payments.

The screenshot shows the MySQL Workbench interface with a query editor window titled "Query 1". The code entered is:

```
681     (8, 8, '2023-08-14', 1100.00),
682     (9, 9, '2023-09-09', 80.00),
683     (10, 10, '2023-10-25', 1500.00);

684
685
686 • UPDATE Vechicles
687     SET dailyRate = 68.00
688     WHERE make = 'Mercedes';
689
690
691 • DELETE FROM Payment
692     WHERE leaseID IN (SELECT leaseID FROM Lease WHERE customer_id = 3);
693
694 • DELETE FROM Lease
695     WHERE customer_id = 3;
696
697 • DELETE FROM Customers
698     WHERE customer_id = 3;
```

The output pane shows the results of the delete queries:

Action	Time	Message	Duration / Fetch
DELETE FROM Payment WHERE leaseID IN (SELECT leaseID FROM Lease WHERE customer_id = 3)	1 21:46:31	3 row(s) affected	0.000 sec
DELETE FROM Lease WHERE customer_id = 3	2 21:46:31	3 row(s) affected	0.000 sec
DELETE FROM Customers WHERE customer_id = 3	3 21:46:31	1 row(s) affected	0.000 sec

### Q3. Rename the "paymentDate" column in the Payment table to "transactionDate".

The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Navigator:** MANAGEMENT, INSTANCE, PERFORMANCE.
- Query Editor:** Shows the following SQL script:

```
689 • DELETE FROM Payment
690 WHERE leaseID IN (SELECT leaseID FROM Lease WHERE customer_id = 3);
692
693 • DELETE FROM Lease
694 WHERE customer_id = 3;
695
696 • DELETE FROM Customers
697 WHERE customer_id = 3;
698
699 • ALTER TABLE Payment
700 RENAME COLUMN paymentDate to transactionDate;
```
- Output Window:** Action Output table showing the results of the executed statements:| # | Time | Action | Message | Duration / Fetch |
| --- | --- | --- | --- | --- |
| 1 | 21:46:31 | DELETE FROM Payment WHERE leaseID IN (SELECT leaseID FROM Lease WHERE customer\_id = 3) | 3 row(s) affected | 0.000 sec |
| 2 | 21:46:31 | DELETE FROM Lease WHERE customer\_id = 3 | 3 row(s) affected | 0.000 sec |
| 3 | 21:46:31 | DELETE FROM Customers WHERE customer\_id = 3 | 1 row(s) affected | 0.000 sec |
| 4 | 21:50:39 | ALTER TABLE Payment RENAME COLUMN paymentDate to transactionDate | 0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0 | 0.031 sec |
- System Bar:** Shows the taskbar with various application icons and the system clock indicating 9:55 PM on 12/10/2023.

### Q4. Find a specific customer by email.

The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Navigator:** MANAGEMENT, INSTANCE, PERFORMANCE.
- Query Editor:** Shows the following SQL script:

```
696 • DELETE FROM Customers
697 WHERE customer_id = 3;
698
699 • ALTER TABLE Payment
700 RENAME COLUMN paymentDate to transactionDate;
```
- Result Grid:** Shows the result of the SELECT query:

customer_id	first_name	last_name	email	phone
8	Emma	Wilson	emma@example.com	5554321098
- Output Window:** Action Output table showing the results of the executed statements:| # | Time | Action | Message | Duration / Fetch |
| --- | --- | --- | --- | --- |
| 2 | 21:46:31 | DELETE FROM Lease WHERE customer\_id = 3 | 3 row(s) affected | 0.000 sec |
| 3 | 21:46:31 | DELETE FROM Customers WHERE customer\_id = 3 | 1 row(s) affected | 0.000 sec |
| 4 | 21:50:39 | ALTER TABLE Payment RENAME COLUMN paymentDate to transactionDate | 0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0 | 0.031 sec |
| 5 | 21:54:06 | SELECT \* FROM Customers WHERE email = 'robert@example.com' LIMIT 0, 1000 | 0 row(s) returned | 0.000 sec / 0.000 sec |
| 6 | 21:55:20 | SELECT \* FROM Customers WHERE email = 'emma@example.com' LIMIT 0, 1000 | 1 row(s) returned | 0.000 sec / 0.000 sec |
- System Bar:** Shows the taskbar with various application icons and the system clock indicating 9:55 PM on 12/10/2023.

## Q5. Get active leases for a specific customer.

The screenshot shows the MySQL Workbench interface with a query editor and a results grid. The query is:

```
700 RENAME COLUMN paymentDate TO transactionDate;
701
702 • SELECT *
703   FROM Customers
704   WHERE email = 'emma@example.com';
705
706 • SELECT *
```

The results grid shows one row of data:

leaseID	vehicleID	customer_id	startDate	endDate	ltype
4	4	4	2023-04-20	2023-04-30	Monthly

Below the results grid is a timeline showing three actions:

#	Time	Action	Message	Duration / Fetch
1	21:57:24	SELECT leaseID, vehicleID, startdate, endDate FROM Lease WHERE customer_id = 3 LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec
2	21:57:58	SELECT * FROM Lease WHERE customer_id = 3 LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec
3	21:58:12	SELECT * FROM Lease WHERE customer_id = 4 LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

## Q6. Find all payments made by a customer with a specific phone number

The screenshot shows the MySQL Workbench interface with a query editor and a results grid. The query is:

```
1055 WHERE YEAR(transactionDate) = 2023;
1056
1057
1058 • SELECT p.paymentID, p.leaseID, p.amount, c.customer_id, c.first_name, c.last_name, c.email, c.phone
1059   FROM Payment p
1060   JOIN Lease l ON p.leaseID = l.leaseID
1061   JOIN Customers c ON l.customer_id = c.customer_id
1062   WHERE c.phone = 5551234567;
1063
```

The results grid shows one row of data:

paymentID	leaseID	amount	customer_id	first_name	last_name	email	phone
2	2	1000	2	Jane	Smith	janesmith@example.com	5551234567

Below the results grid is a timeline showing five actions:

#	Time	Action	Message	Duration / Fetch
2	14:18:02	SELECT s.shelterID AS ShelterID, s.sh_name AS ShelterName, s.location AS ShelterLocation, e.eventID AS EventID, e.e...	9 row(s) returned	0.000 sec / 0.000 sec
3	14:39:57	use CarRentalSystem	0 row(s) affected	0.000 sec
4	14:40:10	SELECT p.paymentID, p.leaseID, p.paymentDate, p.amount, c.customer_id, c.first_name, c.last_name, c.email, c...	Error Code: 1054. Unknown column 'p.paymentDate' in field list'	0.015 sec
5	14:40:47	SELECT p.paymentID, p.leaseID, p.amount, c.customer_id, c.first_name, c.last_name, c.email, c...	1 row(s) returned	0.016 sec / 0.000 sec

## Q7. Calculate the average daily rate of all available cars.

The screenshot shows the MySQL Workbench interface with the following details:

- Query Editor (Query 1):** Contains the following SQL code:

```
704 WHERE email = 'emma@example.com';
705
706 • SELECT *
707   FROM Lease
708 WHERE customer_id = 4;
709
710 • SELECT AVG(dailyRate) AS averageDailyRate;
```
- Result Grid:** Shows the output of the last query:

averageDailyRate
53.714285714285715
- Output Window (Result 52):** Displays the action history with the following entries:

#	Time	Action	Message	Duration / Fetch
1	21:57:24	SELECT leaseID, vehicleID, startDate, endDate FROM Lease WHERE customer_id = 3 LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec
2	21:57:58	SELECT * FROM Lease WHERE customer_id = 3 LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec
3	21:58:12	SELECT * FROM Lease WHERE customer_id = 4 LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
4	22:00:07	SELECT AVG(dailyRate) AS averageDailyRate FROM Vehicles WHERE vstatus = 1 LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

## Q8. Find the car with the highest daily rate.

The screenshot shows the MySQL Workbench interface with the following details:

- Query Editor (Query 1):** Contains the following SQL code:

```
715 FROM Vehicles
716 ORDER BY dailyRate DESC
717 LIMIT 1;
718
719
720
721
```
- Result Grid:** Shows the output of the last query:

vehicleID	make	model	vyear	dailyRate	vstatus	passengerCapacity	engineCapacity
8	Mercedes	C-Class	2022	68	1	8	2599
- Output Window (Vehicles 53):** Displays the action history with the following entries:

#	Time	Action	Message	Duration / Fetch
1	21:57:24	SELECT leaseID, vehicleID, startDate, endDate FROM Lease WHERE customer_id = 3 LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec
2	21:57:58	SELECT * FROM Lease WHERE customer_id = 3 LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec
3	21:58:12	SELECT * FROM Lease WHERE customer_id = 4 LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
4	22:00:07	SELECT AVG(dailyRate) AS averageDailyRate FROM Vehicles WHERE vstatus = 1 LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
5	22:02:06	SELECT * FROM Vehicles ORDER BY dailyRate DESC LIMIT 1	1 row(s) returned	0.000 sec / 0.000 sec

**Q9. Retrieve all cars leased by a specific customer.**

**Q10. Find the details of the most recent lease.**

The screenshot shows the MySQL Workbench interface with a query editor and a results grid. The query is:

```
713 • 
714 •     SELECT *
715 •         FROM Vechicles
716 •     ORDER BY dailyRate DESC
717 •     LIMIT 1;
718 • 
719 •     SELECT *
```

The results grid shows one row of data:

leaseID	vehicleID	customer_id	startDate	endDate	ltype
10	10	10	2023-10-10	2023-10-31	Monthly

Below the results grid, the 'Lease 54' pane shows the execution history:

Action	Time	Message	Duration / Fetch
2 21:57:58 SELECT * FROM Lease WHERE customer_id = 3 LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec	
3 21:58:12 SELECT * FROM Lease WHERE customer_id = 4 LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec	
4 22:00:07 SELECT AVG(dailyRate) AS averageDailyRate FROM Vechicles WHERE status = 1 LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec	
5 22:02:46 SELECT * FROM Vechicles ORDER BY dailyRate DESC LIMIT 1	1 row(s) returned	0.000 sec / 0.000 sec	
6 22:05:04 SELECT * FROM Lease ORDER BY endDate DESC LIMIT 1	1 row(s) returned	0.000 sec / 0.000 sec	

**Q11. List all payments made in the year 2023.**

The screenshot shows the MySQL Workbench interface with a query editor and a results grid. The query is:

```
722     LIMIT 1;
723
724
725 •     SELECT *
726 •         FROM Payment
727 •     WHERE YEAR(transactionDate) = 2023;
728
```

The results grid shows ten rows of data:

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

Below the results grid, the 'Payment 55' pane shows the execution history:

Action	Time	Message	Duration / Fetch
1 22:06:20 SELECT * FROM Payments WHERE YEAR(paymentDate) = 2023 LIMIT 0, 1000	Error Code: 1146. Table 'carentalsystem.payments' doesn't exist	0.000 sec	
2 22:06:40 SELECT * FROM Payments WHERE YEAR(transactionDate) = 2023 LIMIT 0, 1000	Error Code: 1146. Table 'carentalsystem.payments' doesn't exist	0.000 sec	
3 22:06:56 SELECT * FROM Payment WHERE YEAR(transactionDate) = 2023 LIMIT 0, 1000	7 row(s) returned	0.047 sec / 0.000 sec	

## Q12. Retrieve customers who have not made any payments.

The screenshot shows the MySQL Workbench interface with a query editor and a results grid. The query retrieves customer details and joins with lease and payment tables to filter out customers with no payments.

```
SELECT c.customer_id, c.first_name, c.last_name, c.email, c.phone
FROM Customers c
LEFT JOIN Lease l ON c.customer_id = l.customer_id
LEFT JOIN Payment p ON l.leaseID = p.leaseID
WHERE p.paymentID IS NULL;
```

customer_id	first_name	last_name	email	phone
6	Laura	Hall	laura@example.com	5552345678
9	William	Taylor	william@example.com	5553216547

The results show two customers: Laura Hall and William Taylor, both of whom have no payment records.

## Q13. Retrieve Car Details and Their Total Payments.

The screenshot shows the MySQL Workbench interface with a query editor and a results grid. The query retrieves vehicle details and joins with lease and payment tables to calculate the total payments for each vehicle.

```
SELECT v.vehcileID, v.make, v.model, v.vyear, v.dailyRate, v.vstatus, v.passengerCapacity, v.engineCapacity, SUM(p.amount) AS TotalPayments
FROM Vechicles v
JOIN Lease l ON v.vehcileID = l.vehcileID
JOIN Payment p ON l.leaseID = p.leaseID
GROUP BY v.vehcileID, v.make, v.model, v.vyear, v.dailyRate, v.vstatus, v.passengerCapacity, v.engineCapacity;
```

vehcileID	make	model	vyear	dailyRate	vstatus	passengerCapacity	engineCapacity	TotalPayments
1	Toyota	Camry	2022	50	1	4	1450	200
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

The results show car details and their respective total payments, such as a Toyota Camry with a total payment of 200 and a Mercedes C-Class with a total payment of 1100.

## Q14. Calculate Total Payments for Each Customer.

The screenshot shows the MySQL Workbench interface with a query editor window titled "Query 1". The SQL code is:

```
L077
L078
L079 • SELECT c.* , SUM(p.amount) AS TotalPayments
L080   FROM Customers c
L081   LEFT JOIN Lease l ON c.customer_id = l.customer_id
L082   LEFT JOIN Payment p ON l.leaseID = p.leaseID
L083   GROUP BY c.customer_id, c.first_name, c.last_name, c.email, c.phone;
L084
L085
```

The results grid shows the following data:

customer_id	first_name	last_name	email	phone	TotalPayments
1	John	Doe	john.doe@example.com	55555555555	200
2	Jane	Smith	jane.smith@example.com	5551234567	1000
4	Sarah	Brown	sarah@example.com	5554567890	900
5	David	Lee	david@example.com	5559876543	800
6	Laura	Hall	laura@example.com	5552345678	700
7	Michael	Davis	michael@example.com	5559765432	600
8	Emma	Wilson	emma@example.com	5554321098	1100
9	William	Taylor	william@example.com	5552126547	1000
10	Olivia	Adams	olivia@example.com	5557654321	1500

The output pane shows the executed query and its duration:

```
1 14:48:46 SELECT c.* , SUM(p.amount) AS TotalPayments FROM Customers c LEFT JOIN Lease l ON c.customer_id = l.customer_id LEFT JOIN Payment p ON l.leaseID = p.leaseID GROUP BY c.customer_id, c.first_name, c.last_name, c.email, c.phone; 9 row(s) returned
```

Duration / Fetch: 0.000 sec / 0.000 sec

## Q15. List Car Details for Each Lease.

The screenshot shows the MySQL Workbench interface with a query editor window titled "Query 1". The SQL code is:

```
L083
L084
L085 • SELECT l.* , v.*
L086   FROM Lease l
L087   JOIN Vehicles v ON l.vehicleID = v.vehicleID;
```

The results grid shows the following data:

leaseID	vehicleID	customer_id	startDate	endDate	ltype	vehicledID	make	model	vyear	dailyRate	vstatus	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	46	1	7	1500
4	4	4	2023-04-20	2023-04-30	Monthly	4	Nissan	Altima	2023	52	1	7	1200
5	5	5	2023-05-20	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

The output pane shows the executed query and its duration:

```
1 14:50:19 SELECT l.* , v.* FROM Lease l JOIN Vehicles v ON l.vehicleID = v.vehicleID LIMIT 0, 1000
7 row(s) returned
```

Duration / Fetch: 0.000 sec / 0.000 sec

## Q16. Retrieve Details of Active Leases with Customer and Car Information.

The screenshot shows the MySQL Workbench interface with the following query executed:

```
L086    FROM Lease l
L087      JOIN Vechicles v ON l.vehcileID = v.vehcileID;
L088
L089 •  SELECT l.*, c.*, v.*
L090    FROM Lease l
L091      JOIN Customers c ON l.customer_id = c.customer_id
L092      JOIN Vechicles v ON l.vehcileID = v.vehcileID
L093      WHERE l.endDate >= '2023-06-30';
L094
```

The results grid displays the following data:

leaseID	vehcileID	customer_id	startDate	endDate	ltype	customer_id	first_name	last_name	email	phone	vehcileID	make	model	vyear	dailyRa
7	7	7	2023-07-01	2023-07-10	Daily	7	Michael	Davis	michael@example.com	5558765432	7	BMW	3 Series	2023	60
8	8	8	2023-08-12	2023-08-15	Monthly	8	Emma	Wilson	emma@example.com	5554321098	8	Mercedes	C-Class	2022	68
10	10	10	2023-10-10	2023-10-31	Monthly	10	Olivia	Adams	olivia@example.com	5557654321	10	Lexus	ES	2023	54

The output pane shows the following log entries:

#	Time	Action	Message	Duration / Fetch
1	14:53:32	SELECT l.*, c.*, v.*	FROM Lease l JOIN Customers c ON l.customer_id = c.customer_id JOIN Vechicles v ON l.vehcileID = v.vehcileID WHERE l.endDate >= '2023-06-30'; 0 row(s) returned	0.000 sec / 0.000 sec
2	14:54:55	SELECT l.*, c.*, v.*	FROM Lease l JOIN Customers c ON l.customer_id = c.customer_id JOIN Vechicles v ON l.vehcileID = v.vehcileID WHERE l.endDate >= '2023-06-30'; 3 row(s) returned	0.016 sec / 0.000 sec

## Q17. Find the Customer Who Has Spent the Most on Leases.

The screenshot shows the MySQL Workbench interface with the following query executed:

```
L091      JOIN Customers c ON l.customer_id = c.customer_id
L092      JOIN Vechicles v ON l.vehcileID = v.vehcileID
L093      WHERE l.endDate >= '2023-06-30';
L094
L095 •  SELECT c.* , SUM(p.amount) AS TotalSpentOnLeases
L096   FROM Customers c
L097   LEFT JOIN Lease l ON c.customer_id = l.customer_id
L098   LEFT JOIN Payment p ON l.leaseID = p.leaseID
```

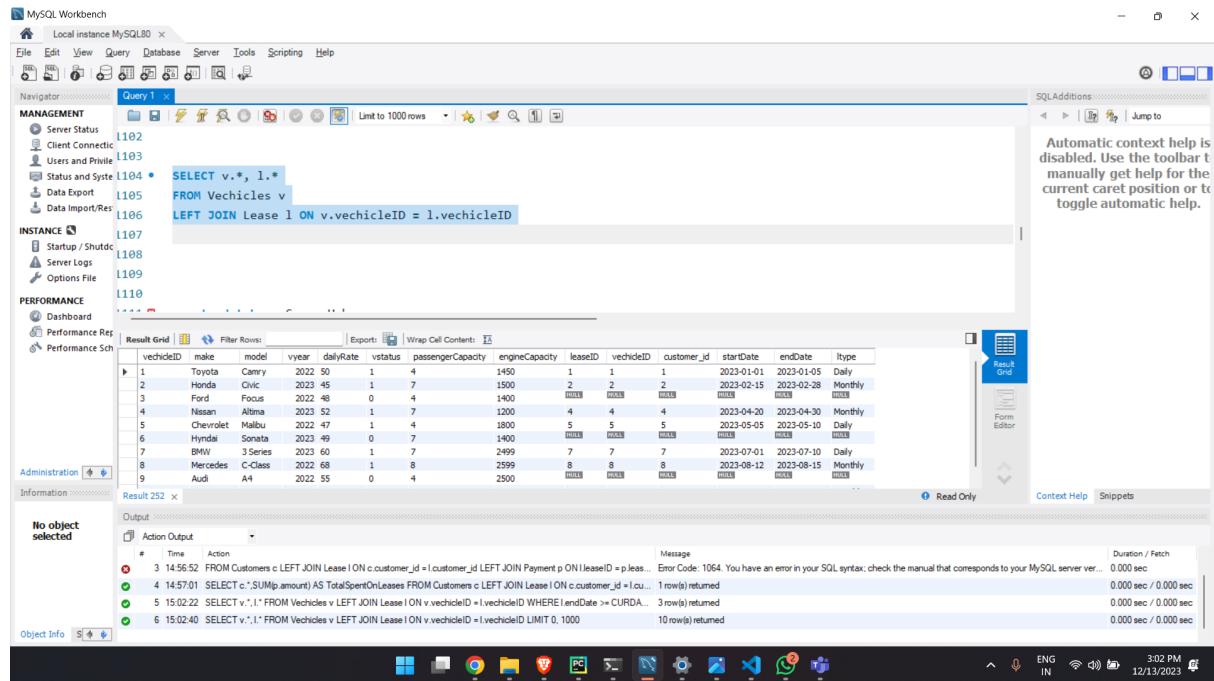
The results grid displays the following data:

customer_id	first_name	last_name	email	phone	TotalSpentOnLeases
10	Olivia	Adams	olivia@example.com	5557654321	1500

The output pane shows the following log entries:

#	Time	Action	Message	Duration / Fetch
1	14:53:32	SELECT l.*, c.*, v.*	FROM Lease l JOIN Customers c ON l.customer_id = c.customer_id JOIN Vechicles v ON l.vehcileID = v.vehcileID WHERE l.endDate >= '2023-06-30'; 0 row(s) returned	0.000 sec / 0.000 sec
2	14:54:55	SELECT l.*, c.*, v.*	FROM Lease l JOIN Customers c ON l.customer_id = c.customer_id JOIN Vechicles v ON l.vehcileID = v.vehcileID WHERE l.endDate >= '2023-06-30'; 3 row(s) returned	0.016 sec / 0.000 sec
3	14:56:52	FROM Customers c LEFT JOIN Lease l ON c.customer_id = l.customer_id LEFT JOIN Payment p ON l.leaseID = p.leaseID	Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version... 0.000 sec	0.000 sec
4	14:57:01	SELECT c.* , SUM(p.amount) AS TotalSpentOnLeases	FROM Customers c LEFT JOIN Lease l ON c.customer_id = l.customer_id WHERE l.endDate >= '2023-06-30'; 1 row(s) returned	0.000 sec / 0.000 sec

## Q18. List All Cars with Their Current Lease Information.



The screenshot shows the MySQL Workbench interface with a query editor and results grid. The query is:

```
SELECT v.* , l.*  
FROM Vechicles v  
LEFT JOIN Lease l ON v.vechicleID = l.vechicleID
```

The results grid displays 252 rows of data from the Vechicles and Lease tables. The columns include vehicleID, make, model, vyear, dailyRate, vstatus, passengerCapacity, engineCapacity, leaseID, vechicleID, customer\_id, startDate, endDate, and ltype. The data shows various car models like Toyota Camry, Honda Civic, Ford Focus, Nissan Altima, Chevrolet Malibu, Hyundai Sonata, BMW 3 Series, Mercedes C-Class, and Audi A4, each with their respective details and lease information.

The output pane shows the following log entries:

- # 3 14:56:52 FROM Customers c LEFT JOIN Lease l ON c.customer\_id = l.customer\_id LEFT JOIN Payment p ON l.leaseID = p.leaseID Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'FROM Customers c LEFT JOIN Lease l ON c.customer\_id = l.customer\_id LEFT JOIN Paym' Duration / Fetch 0.000 sec
- 4 14:57:01 SELECT c.`SUM(amount)` AS TotalSpentOnLeases FROM Customers c LEFT JOIN Lease l ON c.customer\_id = l.customer\_id WHERE l.endDate >= CURDATE() 1 row(s) returned Duration / Fetch 0.000 sec / 0.000 sec
- 5 15:02:22 SELECT v.\* , l.\* FROM Vechicles v LEFT JOIN Lease l ON v.vechicleID = l.vechicleID WHERE l.endDate >= CURDATE() 3 row(s) returned Duration / Fetch 0.000 sec / 0.000 sec
- 6 15:02:40 SELECT v.\* , l.\* FROM Vechicles v LEFT JOIN Lease l ON v.vechicleID = l.vechicleID LIMIT 0, 1000 10 row(s) returned Duration / Fetch 0.000 sec / 0.000 sec