

Coding Challenge - Car Rental System – SQL

Work by –

PAVITHRA B

Pavithrabala2203@gmail.com

```
CREATE DATABASE car_rental_system;  
USE car_rental_system;
```

-- Creating vehicle table

```
CREATE TABLE Vehicle(  
    vehicleID INT PRIMARY KEY,  
    make VARCHAR(50) NOT NULL,  
    model VARCHAR(50) NOT NULL,  
    year INT NOT NULL CHECK (year >= 2000),  
    dailyRate DECIMAL(10,2) NOT NULL CHECK (dailyRate > 0),  
    status INT CHECK (status=0 OR status=1) NOT NULL,  
    passengerCapacity INT NOT NULL CHECK (passengerCapacity > 0),  
    engineCapacity INT NOT NULL CHECK (engineCapacity > 0)  
);
```

-- Creating Customer Table

```
CREATE TABLE Customer (  
    customerID INT PRIMARY KEY ,  
    firstName VARCHAR(50) NOT NULL,  
    lastName VARCHAR(50) NOT NULL,  
    email VARCHAR(100) UNIQUE NOT NULL,  
    phoneNumber VARCHAR(15) UNIQUE NOT NULL  
);
```

-- Creating Lease Table

```
CREATE TABLE Lease (
    leaseID INT PRIMARY KEY ,
    vehicleID INT NOT NULL,
    customerID INT NOT NULL,
    startDate DATE NOT NULL,
    endDate DATE NOT NULL ,
    type ENUM('DailyLease', 'MonthlyLease') NOT NULL,
    FOREIGN KEY (vehicleID) REFERENCES Vehicle(vehicleID) ON DELETE CASCADE,
    FOREIGN KEY (customerID) REFERENCES Customer(customerID) ON DELETE CASCADE
);
```

-- Creating Payment Table

```
CREATE TABLE Payment (
    paymentID INT PRIMARY KEY ,
    leaseID INT NOT NULL,
    paymentDate DATE NOT NULL ,
    amount DECIMAL(10,2) NOT NULL CHECK (amount > 0),
    FOREIGN KEY (leaseID) REFERENCES Lease(leaseID) ON DELETE CASCADE
);
```

-- Inserting values in Vehicle table

```
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);
```

-- Inserting values in Customer table

```
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');
```

-- Inserting values in Lease table

```
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');
```

-- Inserting values in Payment table

```
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.

UPDATE Vehicle

SET dailyRate = 68.00

WHERE vehicleID=8;

The screenshot shows the MySQL Workbench interface. In the SQL editor pane, the following SQL code is written:

```
113
114
115
116
117
118 -- 1. Update the daily rate for a Mercedes car to 68.
119 • UPDATE Vehicle
120 SET dailyRate = 68.00
121 WHERE vehicleID=8;
122
123
124
125
126
127
128
129
```

The code is annotated with a comment -- 1. Update the daily rate for a Mercedes car to 68. The line number 119 is highlighted with a blue dot, indicating it is the current line of execution. The WHERE clause specifies vehicleID=8.

In the Output pane, the results of the executed statements are shown:

#	Time	Action	Message	Duration / Fetch
43	13:26:18	SELECT c.customerID, c.firstName, c.lastName, SUM(p.amount) AS totalPayments FROM Customer c JOIN Lease l ON c.customerID = l.customerID JOIN Vehicle v ON l.vehicleID = v.vehicleID WHERE v.model = 'Mercedes' AND l.startDate > '2023-01-01' AND l.endDate < '2023-02-01'	8 row(s) returned	0.000 sec / 0.000 sec
44	13:27:17	SELECT l.leaseID, v.make, v.year, l.startDate, l.endDate, l.duration, l.totalPayments, c.customerID, c.firstName, c.lastName, SUM(p.amount) AS totalSpent FROM Lease l JOIN Vehicle v ON l.vehicleID = v.vehicleID JOIN Customer c ON l.customerID = c.customerID JOIN Payment p ON l.leaseID = p.leaseID WHERE v.model = 'Mercedes' AND l.startDate > '2023-01-01' AND l.endDate < '2023-02-01' ORDER BY l.startDate	10 row(s) returned	0.000 sec / 0.000 sec
45	13:27:37	SELECT * FROM Lease l JOIN Vehicle v ON l.vehicleID = v.vehicleID LIMIT 0, 1000	10 row(s) returned	0.016 sec / 0.000 sec
46	13:29:07	SELECT l.leaseID, c.firstName, c.lastName, v.make, v.model, l.startDate, l.endDate, l.duration, l.totalPayments, c.customerID, c.firstName, c.lastName, SUM(p.amount) AS totalSpent FROM Lease l JOIN Customer c ON l.customerID = c.customerID JOIN Vehicle v ON l.vehicleID = v.vehicleID JOIN Payment p ON l.leaseID = p.leaseID WHERE v.model = 'Mercedes' AND l.startDate > '2023-01-01' AND l.endDate < '2023-02-01' ORDER BY l.startDate	0 row(s) returned	0.000 sec / 0.000 sec
47	13:29:52	SELECT c.customerID, c.firstName, c.lastName, SUM(p.amount) AS totalSpent FROM Customer c JOIN Lease l ON c.customerID = l.customerID JOIN Vehicle v ON l.vehicleID = v.vehicleID WHERE v.model = 'Mercedes' AND l.startDate > '2023-01-01' AND l.endDate < '2023-02-01'	1 row(s) returned	0.016 sec / 0.000 sec
48	13:30:41	SELECT v.vehicleID, v.make, v.model, v.year, l.startDate, l.endDate, c.firstName, c.lastName FROM Vehicle v JOIN Lease l ON v.vehicleID = l.vehicleID JOIN Customer c ON l.customerID = c.customerID WHERE v.model = 'Mercedes' AND l.startDate > '2023-01-01' AND l.endDate < '2023-02-01'	12 row(s) returned	0.000 sec / 0.000 sec
49	13:41:50	DELETE FROM Customer WHERE customerID = 2	1 row(s) affected	0.000 sec
50	13:50:44	UPDATE Vehicle SET dailyRate = 68.00 WHERE vehicleID=8	0 row(s) affected Rows matched: 1 Changed: 0 Warnings: 0	0.000 sec
51	13:50:58	UPDATE Vehicle SET dailyRate = 68.00 WHERE vehicleID=8	0 row(s) affected Rows matched: 1 Changed: 0 Warnings: 0	0.000 sec

The output shows that the update statement (line 50) was successful, changing the daily rate for vehicle ID 8 to 68.00. The status bar at the bottom indicates the session is connected to "Partly sunny" and the system time is 1:51 PM on 4/2/2025.

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

DELETE FROM Customer WHERE customerID = 2;

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

- Navigator:** Shows the database schema with the `ticketbookingSystem` database selected, containing tables like `asset_allocations`, `assets`, `booking`, `customer`, `employees`, `event`, and `maintenance_records`.
- SQL Editor:** Contains the SQL code:

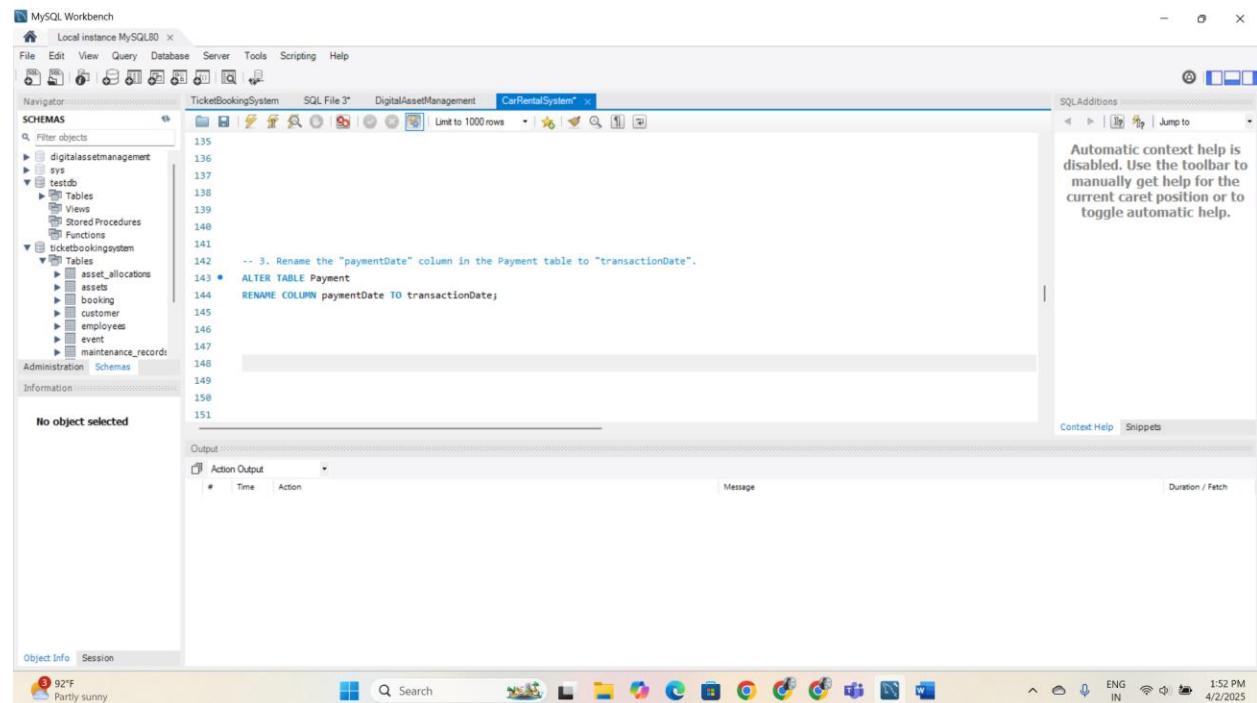
```
125
126
127
128
129
130
131 -- 2. Delete a specific customer and all associated leases and payments.
132 • DELETE FROM Customer WHERE customerID = 2;
133
134
135
136
137
138
139
140
141
```
- Output:** Displays the execution log:

#	Time	Action	Message	Duration / Fetch
44	13:27:17	SELECT l.leaseID, v.make, v.model, v.year, l.startDate, l.endDate FROM Lease l JOIN Vehicle v ON l.vehicleID = v.vehicleID	10 row(s) returned	0.000 sec / 0.000 sec
45	13:27:37	SELECT * FROM Lease l JOIN Vehicle v ON l.vehicleID = v.vehicleID LIMIT 0, 1000	10 row(s) returned	0.016 sec / 0.000 sec
46	13:29:07	SELECT l.leaseID, c.firstName, c.lastName, v.make, v.model, l.startDate, l.endDate FROM Lease l JOIN Cust...	0 row(s) returned	0.000 sec / 0.000 sec
47	13:29:52	SELECT c.customerID, c.firstName, c.lastName, SUM(lp.amount) AS totalSpent FROM Customer c JOIN Leas...	1 row(s) returned	0.016 sec / 0.000 sec
48	13:30:41	SELECT v.vehicleID, v.make, v.model, v.year, l.startDate, l.endDate, c.firstName, c.lastName FROM Vehicle ...	12 row(s) returned	0.000 sec / 0.000 sec
49	13:41:50	DELETE FROM Customer WHERE customerID = 2	1 row(s) affected	0.000 sec
50	13:50:44	UPDATE Vehicle SET dailyRate = 68.00 WHERE vehicleID=8	0 row(s) affected Rows matched: 1 Changed: 0 Warnings: 0	0.000 sec
51	13:50:58	UPDATE Vehicle SET dailyRate = 68.00 WHERE vehicleID=8	0 row(s) affected Rows matched: 1 Changed: 0 Warnings: 0	0.000 sec
52	13:51:26	DELETE FROM Customer WHERE customerID = 2	0 row(s) affected	0.000 sec

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

ALTER TABLE Payment

RENAME COLUMN paymentDate TO transactionDate;



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

- Title Bar:** MySQL Workbench - Local instance MySQL80
- Toolbar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help
- Navigator:** Shows SCHEMAS (digitalassetmanagement, sys, testdb) and TABLES (Tables, Views, Stored Procedures, Functions) under ticketbookingystem.
- SQL Editor:** Contains the following script:

```
135
136
137
138
139
140
141
142 -- 3. Rename the "paymentDate" column in the Payment table to "transactionDate".
143 * ALTER TABLE Payment
144   RENAME COLUMN paymentDate TO transactionDate;
145
146
147
148
149
150
151
```
- Output:** Action Output tab is selected, showing columns for #, Time, and Action. The message area is empty.
- System Tray:** Shows weather (92°F, Partly sunny), system icons, and a taskbar with various application icons.
- Bottom Bar:** Shows the date and time (4/2/2025, 1:52 PM).

-- 4. Find a specific customer by email.

```
SELECT * FROM Customer
```

```
WHERE email = 'robert@example.com';
```

The screenshot shows the MySQL Workbench interface. In the central SQL editor pane, the following code is displayed:

```
153
154
155
156 -- 4. Find a specific customer by email.
157 • SELECT * FROM Customer
158 WHERE email = 'robert@example.com'
159
160
161
```

The code is annotated with a bullet point at line 157 indicating it is the current step. Below the editor is the Result Grid, which displays the following data:

customerID	firstName	lastName	email	phoneNumber
3	Robert	Johnson	robert@example.com	555-789-1234

At the bottom of the window, the Output pane shows the execution details:

#	Time	Action	Message	Duration / Fetch
1	13:52:55	SELECT * FROM Customer WHERE email = 'robert@example.com' LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

The system tray at the bottom of the screen shows the date and time as 4/2/2025 1:53 PM.

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

```
SELECT endDate AS ACTIVE_LEASES FROM Lease  
WHERE customerID = 2 ;
```

The screenshot shows the MySQL Workbench interface. The SQL editor tab contains the following code:

```
167  
168  
169  
170 -- 5.Get active leases for a specific customer.  
171 • SELECT endDate AS ACTIVE_LEASES FROM Lease  
172 WHERE customerID = 2 ;  
173  
174  
175  
176
```

The results pane shows a single row of data:

ACTIVE_LEASES
2023-04-02 12:00:00

The output pane shows the following log entries:

#	Time	Action	Message	Duration / Fetch
1	13:52:55	SELECT * FROM Customer WHERE email = 'robert@example.com' LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
2	13:53:27	SELECT endDate AS ACTIVE_LEASES FROM Lease WHERE customerID = 2 LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec

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

```
SELECT p.* FROM Payment p  
JOIN Lease l ON p.leaseID = l.leaseID  
JOIN Customer c ON l.customerID = c.customerID  
WHERE c.phoneNumber = '555-123-4567';
```

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

- Navigator:** Shows the database schema with tables like asset_allocations, assets, booking, customer, employees, event, and maintenance_records.
- SQL Editor:** Contains the SQL query for finding payments by a specific customer.
- Result Grid:** Displays the results of the query, which are empty in this case.
- Action Output:** Shows the log of actions taken by the system, including the execution of the query.
- System Tray:** Shows the Windows taskbar with various application icons and the date/time (1:54 PM, 4/2/2025).

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

```
SELECT AVG(dailyRate) AS averageDailyRate
```

```
FROM Vehicle
```

```
WHERE status = 1;
```

The screenshot shows the MySQL Workbench interface. The SQL editor tab contains the following code:

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

The results pane shows a single row of output:

averageDailyRate
53.71426

The status bar at the bottom right indicates the session details: 92°F, Partly sunny, ENG IN, 1:54 PM, 4/2/2025.

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

```
SELECT MAX(dailyRate) AS highestDailyRate  
FROM Vehicle ;
```

The screenshot shows the MySQL Workbench interface. The left sidebar displays the database schema with two main schemas: 'digitalassetmanagement' and 'ticketbookingsystem'. The 'ticketbookingsystem' schema contains several tables: asset_allocations, assets, booking, customer, employees, event, and maintenance_records. The central pane shows the SQL editor with the following code:

```
195  
196  
197 -- 8. Find the car with the highest daily rate.  
198 * SELECT MAX(dailyRate) AS highestDailyRate  
199 FROM Vehicle ;  
200  
201  
202  
203  
~n~
```

The result grid shows the output of the query:

highestDailyRate
68.00

The bottom pane shows the 'Output' tab with the following log entries:

#	Time	Action	Message	Duration / Fetch
1	13:52:55	SELECT * FROM Customer WHERE email = 'robert@example.com' LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
2	13:53:27	SELECT endDate AS ACTIVELEASES FROM Lease WHERE customerID = 2 LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec
3	13:54:05	SELECT p.* FROM Payment p JOIN Lease l ON p.leaseID = l.leaseID JOIN Customer c ON l.customerID = c.customerID WHERE status = 1 LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec
4	13:54:34	SELECT AVG(dailyRate) AS averageDailyRate FROM Vehicle WHERE status = 1 LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
5	13:55:15	SELECT MAX(dailyRate) AS highestDailyRate FROM Vehicle LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

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

```
SELECT v.make FROM Vehicle v
```

```
JOIN Lease l ON v.vehicleID = l.vehicleID
```

```
WHERE l.customerID = 2;
```

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

- Schemas:** The current schema is `TicketBookingSystem`.
- Query Editor:** The SQL query is:`284
285
286 -- 9. Retrieve all cars leased by a specific customer.
287 • SELECT v.make FROM Vehicle v
288 JOIN Lease l ON v.vehicleID = l.vehicleID
289 WHERE l.customerID = 2;
290
291
292`
- Result Grid:** The result grid shows a single column named `make` with one row containing the value `make`.
- Action Output:** The log shows the following actions:

#	Time	Action	Message	Duration / Fetch
1	13:52:55	SELECT * FROM Customer WHERE email = 'robert@example.com' LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
2	13:53:27	SELECT endDate AS ACTIVELEASES FROM Lease WHERE customerID = 2 LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec
3	13:54:05	SELECT p.* FROM Payment p JOIN Lease l ON p.leaseID = l.leaseID JOIN Customer c ON l.customerID = c.customerID WHERE status = 1 LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec
4	13:54:34	SELECT AVG(dailyRate) AS averageDailyRate FROM Vehicle WHERE status = 1 LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
5	13:55:15	SELECT MAX(dailyRate) AS highestDailyRate FROM Vehicle LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
6	13:55:44	SELECT v.make FROM Vehicle v JOIN Lease l ON v.vehicleID = l.vehicleID WHERE l.customerID = 2 LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec
- System Bar:** Shows the date and time as 4/2/2025, 1:55 PM.

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

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

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

- Schemas:** The current schema is `TicketBookingSystem`.
- SQL Editor:** The query text is:

```
214  
215 -- 10. Find the details of the most recent lease.  
216 • SELECT *  
217 FROM Lease  
218 WHERE startDate = (SELECT MAX(startDate) FROM Lease);  
219  
220  
221  
222  
~~~
```
- Result Grid:** A table titled "Lease" is displayed with the following data:

leaseID	vehicleID	customerID	startDate	endDate	type
10	10	10	2023-10-10	2023-10-31	MonthlyLease
- Action Output:** A table showing the execution history:

#	Time	Action	Message	Duration / Fetch
1	13:52:55	SELECT * FROM Customer WHERE email = 'robert@example.com' LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
2	13:53:27	SELECT endDate AS ACTIVE_LEASES FROM Lease WHERE customerID = 2 LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec
3	13:54:05	SELECT p.* FROM Payment p JOIN Lease l ON p.leaseID = l.leaseID JOIN Customer c ON l.customerID = c.customerID WHERE status = 1 LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec
4	13:54:34	SELECT AVG(dailyRate) AS averageDailyRate FROM Vehicle WHERE status = 1 LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
5	13:55:15	SELECT MAX(dailyRate) AS highestDailyRate FROM Vehicle LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
6	13:55:44	SELECT v.make FROM Vehicle v JOIN Lease l ON v.vehicleID = l.vehicleID WHERE l.customerID = 2 LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec
7	13:56:09	SELECT * FROM Lease WHERE startDate = (SELECT MAX(startDate) FROM Lease) LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
- System Bar:** Shows the date and time as 4/2/2025, 1:56 PM.

-- 11. List all payments made in the year 2023.

```
SELECT * FROM Payment
```

```
WHERE YEAR(transactionDate) = 2023;
```

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

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Navigator:** Local instance MySQL80, Schemas (digitalassetmanagement, sys, ticketdb, ticketbookingsystem), Tables (asset_allocations, assets, booking, customer, employees, event, maintenance_records).
- SQL Editor:** Current tab is CarRentalSystem*. The SQL code is:

```
220
221
222
223
224 -- 11. List all payments made in the year 2023.
225 • SELECT * FROM Payment
226 WHERE YEAR(transactionDate) = 2023;
227
228
```
- Result Grid:** Shows the results of the query:

paymentID	leaseID	transactionDate	amount
1	1	2023-01-03	200.00
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
- Action Output:** Shows the execution history:

#	Time	Action	Message	Duration / Fetch
1	13:52:55	SELECT * FROM Customer WHERE email = 'robert@example.com' LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
2	13:53:27	SELECT endDate AS ACTIVELEASES FROM Lease WHERE customerID = 2 LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec
3	13:54:05	SELECT p.* FROM Payment p JOIN Lease l ON p.leaseID = l.leaseID JOIN Customer c ON l.customerID = c.customerID WHERE l.endDate > CURDATE() AND l.status = 1 LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec
4	13:54:34	SELECT AVG(dailyRate) AS averageDailyRate FROM Vehicle WHERE status = 1 LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
5	13:55:15	SELECT MAX(dailyRate) AS highestDailyRate FROM Vehicle LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
6	13:55:44	SELECT v.* FROM Vehicle v JOIN Lease l ON v.vehicleID = l.vehicleID WHERE l.customerID = 2 LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec
7	13:56:09	SELECT * FROM Lease WHERE startDate = (SELECT MAX(startDate) FROM Lease) LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
8	13:56:48	SELECT * FROM Payment WHERE YEAR(transactionDate) = 2023 LIMIT 0, 1000	9 row(s) returned	0.000 sec / 0.000 sec
- Session Bar:** Object Info, Session, Weather (92°F, Partly sunny), Search, Taskbar icons, Language (ENG IN), Network icons, Date/Time (1:56 PM, 4/2/2025).

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

```
SELECT c.* FROM Customer c
```

```
LEFT JOIN Lease l ON c.customerID = l.customerID
```

```
LEFT JOIN Payment p ON l.leaseID = p.leaseID
```

```
WHERE p.paymentID IS NULL;
```

The screenshot shows the MySQL Workbench interface. The SQL editor tab contains the following code:

```
229
230 -- 12. Retrieve customers who have not made any payments.
231 • SELECT c.* FROM Customer c
232 LEFT JOIN Lease l ON c.customerID = l.customerID
233 LEFT JOIN Payment p ON l.leaseID = p.leaseID
234 WHERE p.paymentID IS NULL;
235
236
237
```

The Result Grid shows two rows of data:

customerID	firstName	lastName	email	phoneNumber
6	Laura	Hall	laura@example.com	555-234-5678
9	William	Taylor	william@example.com	555-321-6547

The Output tab shows the execution log:

#	Time	Action	Message	Duration / Fetch
1	13:52:55	SELECT * FROM Customer WHERE email = 'robert@example.com' LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
2	13:53:27	SELECT endDate AS ACTIVELEASES FROM Lease WHERE customerID = 2 LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec
3	13:54:05	SELECT p.* FROM Payment p JOIN Lease l ON p.leaseID = l.leaseID JOIN Customer c ON l.customerID = c.customerID WHERE l.customerID = 2 AND p.paymentID IS NULL LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec
4	13:54:34	SELECT AVG(dailyRate) AS averageDailyRate FROM Vehicle WHERE status = 1 LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
5	13:55:15	SELECT MAX(dailyRate) AS highestDailyRate FROM Vehicle LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
6	13:55:44	SELECT v.make FROM Vehicle v JOIN Lease l ON v.vehicleID = l.vehicleID WHERE l.customerID = 2 LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec
7	13:56:09	SELECT * FROM Lease WHERE startDate = (SELECT MAX(startDate) FROM Lease) LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
8	13:56:48	SELECT * FROM Payment WHERE YEAR(transactionDate) = 2023 LIMIT 0, 1000	9 row(s) returned	0.000 sec / 0.000 sec
9	13:57:18	SELECT c.* FROM Customer c LEFT JOIN Lease l ON c.customerID = l.customerID LEFT JOIN Payment p ON l.leaseID = p.leaseID WHERE p.paymentID IS NULL LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec

-- 13. Retrieve Car Details and Their Total Payments.

```
SELECT v.vehicleID, v.make, v.model, 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, v.make, v.model;
```

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

- Navigator:** Shows the database structure with Schemas: digitalassetmanagement, sys, testdb, ticketbookingsystem. Under ticketbookingsystem, there are Tables: asset_allocations, assets, booking, customer, employee, event, maintenance_records.
- SQL Editor:** Contains the SQL query for retrieving car details and their total payments.
- Result Grid:** Displays the results of the query, showing 7 rows of data:

vehicleID	make	model	totalPayments
1	Toyota	Camry	200.00
3	Ford	Focus	155.00
4	Nissan	Altima	2100.00
5	Chevrolet	Hhrbu	60.00
7	BMW	3 Series	40.00

- Action Output:** Shows the execution log with 10 entries, each detailing a query execution step.
- System Status:** Shows the system status bar at the bottom with information like weather (92°F, Partly sunny), system icons, and network status.

-- 14. Calculate Total Payments for Each Customer.

```
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, c.firstName, c.lastName;
```

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

- Schemas:** Local instance MySQL80, ticketbookingSystem
- Tables:** asset_allocations, assets, booking, customer, employee, event, maintenance_records
- Query Editor:** Contains the SQL code for query 14.
- Result Grid:** Displays the output of the query, showing 7 rows of data:

customerID	firstName	lastName	totalPayments
1	John	Doe	200.00
3	Robert	Johnson	1355.00
4	Sarah	Brown	900.00
5	David	Lee	60.00
7	Michael	Davis	40.00

- Action Output:** Shows the execution log with 11 entries, each with a timestamp, action, message, and duration.
- System Status:** Shows weather (92°F, Partly sunny), system icons, and system status (ENG IN, 1:58 PM, 4/2/2025).

-- 15. List Car Details for Each Lease.

```
SELECT l.leaseID, v.make, v.model, v.year, l.startDate, l.endDate  
FROM Lease l  
JOIN Vehicle v ON l.vehicleID = v.vehicleID;
```

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

- Schemas:** The current schema is "TicketBookingSystem".
- SQL Editor:** The query is displayed:

```
253  
254  
255  
256  
257 -- 15. List Car Details for Each Lease.  
258 • SELECT l.leaseID, v.make, v.model, v.year, l.startDate, l.endDate  
259 FROM Lease l  
260 JOIN Vehicle v ON l.vehicleID = v.vehicleID;  
261  
~^
```
- Result Grid:** The result of the query is shown in a grid format:

leaseID	make	model	year	startDate	endDate
1	Toyota	Camry	2022	2023-01-01	2023-01-05
3	Ford	Focus	2022	2023-03-10	2023-03-15
4	Nissan	Altima	2023	2023-04-20	2023-04-30
5	Chevrolet	Malibu	2022	2023-05-05	2023-05-10
6	Nissan	Altima	2023	2023-06-15	2023-06-30
- Action Output:** The log of executed statements is shown:

#	Time	Action	Message	Duration / Fetch
4	13:54:34	SELECT AVG(dailyRate) AS averageDailyRate FROM Vehicle WHERE status = 1 LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
5	13:55:15	SELECT MAX(dailyRate) AS highestDailyRate FROM Vehicle LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
6	13:55:44	SELECT v.make FROM Vehicle v JOIN Lease l ON v.vehicleID = l.vehicleID WHERE l.customerID = 2 LIMIT ...	0 row(s) returned	0.000 sec / 0.000 sec
7	13:56:09	SELECT * FROM Lease WHERE startDate = (SELECT MAX(startDate) FROM Lease) LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
8	13:56:48	SELECT * FROM Payment WHERE YEAR(transactionDate) = 2023 LIMIT 0, 1000	9 row(s) returned	0.000 sec / 0.000 sec
9	13:57:18	SELECT c.* FROM Customer c LEFT JOIN Lease l ON c.customerID = l.customerID LEFT JOIN Payment p ON ...	2 row(s) returned	0.000 sec / 0.000 sec
10	13:57:41	SELECT v.vehicleID, v.make, v.model, SUM(p.amount) AS totalPayments FROM Vehicle v JOIN Lease l ON ...	7 row(s) returned	0.000 sec / 0.000 sec
11	13:58:09	SELECT c.customerID, c.firstName, c.lastName, SUM(p.amount) AS totalPayments FROM Customer c JOIN ...	7 row(s) returned	0.000 sec / 0.000 sec
12	13:58:30	SELECT l.leaseID, v.make, v.model, v.year, l.startDate, l.endDate FROM Lease l JOIN Vehicle v ON l.vehicleID = v.vehicleID	9 row(s) returned	0.000 sec / 0.000 sec
- System Bar:** Shows the date (4/2/2025), time (1:58 PM), and system status (ENG IN).

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

```
SELECT l.leaseID, c.firstName, c.lastName, v.make, v.model, l.startDate, l.endDate  
FROM Lease l  
JOIN Customer c ON l.customerID = c.customerID  
JOIN Vehicle v ON l.vehicleID = v.vehicleID  
WHERE l.endDate >= CURDATE();
```

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

- Schemas:** Local instance MySQL80, TicketBookingSystem, SQL File 3*, DigitalAssetManagement, CarRentalSystem*
- Query Editor:** Contains the SQL code for query 16.
- Result Grid:** Displays the results of the query, showing columns: leaseID, firstName, lastName, make, model, startDate, endDate.
- Output Window:** Shows the execution log with 13 entries, each with a timestamp, action, message, and duration.
- System Bar:** Includes icons for file operations, search, and system status (92°F, partly sunny).
- Bottom Status:** Shows the date (4/2/2025), time (1:59 PM), and language (ENG IN).

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

```
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, c.firstName, c.lastName  
ORDER BY totalSpent DESC  
LIMIT 1;
```

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

- Navigator:** Shows the database schema with the `TicketBookingSystem` database selected. Under `TicketBookingSystem`, there are tables like `customer`, `lease`, and `payment`.
- SQL Editor:** Contains the SQL query from above. The result grid shows one row: customerID 10, firstName Olivia, lastName Adams, and totalSpent 1500.00.
- Output:** Displays the execution log with 14 entries, each showing a query number, time, action, message, and duration.
- System Bar:** Shows the system tray with weather information (92°F, Partly sunny), network status, battery level, and system icons.

#	Time	Action	Message	Duration / Fetch
6	13:55:44	SELECT v.make FROM Vehicle v JOIN Lease l ON vehicleID = l.vehicleID WHERE l.customerID = 2 LIMIT ...	0 row(s) returned	0.000 sec / 0.000 sec
7	13:56:09	SELECT * FROM Lease WHERE startDate = (SELECT MAX(startDate) FROM Lease) LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
8	13:56:48	SELECT * FROM Payment WHERE YEAR(transactionDate) = 2023 LIMIT 0, 1000	9 row(s) returned	0.000 sec / 0.000 sec
9	13:57:18	SELECT c.* FROM Customer c LEFT JOIN Lease l ON c.customerID = l.customerID LEFT JOIN Payment p ...	2 row(s) returned	0.000 sec / 0.000 sec
10	13:57:41	SELECT v.vehicleID, v.make, v.model, SUM(p.amount) AS totalPayments FROM Vehicle v JOIN Lease l ON ...	7 row(s) returned	0.000 sec / 0.000 sec
11	13:58:09	SELECT c.customerID, c.firstName, c.lastName, SUM(p.amount) AS totalPayments FROM Customer c JOIN L...	7 row(s) returned	0.000 sec / 0.000 sec
12	13:58:30	SELECT l.leaseID, v.make, v.model, v.year, l.startDate, l.endDate FROM Lease l JOIN Vehicle v ON l.vehicleID ...	9 row(s) returned	0.000 sec / 0.000 sec
13	13:58:57	SELECT l.leaseID, c.firstName, c.lastName, v.make, v.model, l.startDate, l.endDate FROM Lease l JOIN Cust...	0 row(s) returned	0.000 sec / 0.000 sec
14	13:59:27	SELECT c.customerID, c.firstName, c.lastName, SUM(p.amount) AS totalSpent FROM Customer c JOIN Lease l ...	1 row(s) returned	0.000 sec / 0.000 sec

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

```
SELECT v.vehicleID, v.make, v.model, v.year, l.startDate, l.endDate, c.firstName, c.lastName  
FROM Vehicle v  
LEFT JOIN Lease l ON v.vehicleID = l.vehicleID  
LEFT JOIN Customer c ON l.customerID = c.customerID;
```

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

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Toolbar:** Standard MySQL icons for connection, schema, table, view, stored procedure, function, and table structure.
- Navigator:** Shows the database structure under "Schemas".
 - testdb: Tables (asset_allocations, assets, booking, customer, employees, event, maintenance_records)
 - ticketbookingystem: Tables (asset_allocations, assets, booking, customer, employees, event, maintenance_records)
- SQL Editor:** Contains the SQL query for listing all cars with their current lease information.
- Result Grid:** Displays the query results in a tabular format.

vehicleID	make	model	year	startDate	endDate	firstName	lastName
1	Toyota	Camry	2022	2023-01-01	2023-01-05	John	Doe
2	Honda	Civic	2023	2023-02-01	2023-02-05	Mary	Smith
3	Ford	Focus	2022	2023-03-10	2023-03-15	Robert	Johnson
3	Ford	Focus	2022	2023-09-07	2023-09-10	Robert	Johnson
4	Nissan	Altima	2023	2023-04-20	2023-04-30	Sarah	Brown
- Output Tab:** Shows the "Action Output" section with 15 rows of log entries, each detailing a query execution with time, action, message, and duration/fetch details.
- System Bar:** Includes weather (92°F, Partly sunny), search, and various system icons.
- Bottom Bar:** Language (ENG IN), date (4/2/2025), and time (1:59 PM).