

Creating DB -

```
1 • create database car_rental_system;
2 • use car_rental_system;
```

Creating Vehicle Table -

```
1 • create table Vehicle (
2     vehicleID int primary key not null auto_increment,
3     make varchar(255),
4     model varchar(255),
5     year int,
6     dailyRate decimal(10, 2),
7     status enum('available', 'notavailable'),
8     passengerCapacity int,
9     engineCapacity int
10 );
11
12 • desc Vehicle;
```

Field	Type	Null	Key	Default	Extra
vehicleID	int	NO	PRI	NULL	auto_increment
make	varchar(255)	YES		NULL	
model	varchar(255)	YES		NULL	
year	int	YES		NULL	
dailyRate	decimal(10,2)	YES		NULL	
status	enum('available','notavailable')	YES		NULL	
passengerCapacity	int	YES		NULL	
engineCapacity	int	YES		NULL	

```
1 • insert into vehicle (make, model, year, dailyRate, status, passengerCapacity, engineCapacity)
2 values
3     ('Toyota', 'Camry', 2022, 50.00, 'available', 4, 1450),
4     ('Honda', 'Civic', 2023, 45.00, 'available', 7, 1500),
5     ('Ford', 'Focus', 2022, 48.00, 'notavailable', 4, 1400),
6     ('Nissan', 'Altima', 2023, 52.00, 'available', 7, 1200),
7     ('Chevrolet', 'Malibu', 2022, 47.00, 'available', 4, 1800),
8     ('Hyundai', 'Sonata', 2023, 49.00, 'notavailable', 7, 1400),
9     ('BMW', '3 Series', 2023, 60.00, 'available', 7, 2499),
10    ('Mercedes', 'C-Class', 2022, 58.00, 'available', 8, 2599),
11    ('Audi', 'A4', 2022, 55.00, 'notavailable', 4, 2500),
12    ('Lexus', 'ES', 2023, 54.00, 'available', 4, 2500);
13
```

vehicleID	make	model	year	dailyRate	status	passengerCapacity	engineCapacity
11	Toyota	Camry	2022	50.00	available	4	1450
12	Honda	Civic	2023	45.00	available	7	1500
13	Ford	Focus	2022	48.00	notavailable	4	1400
14	Nissan	Altima	2023	52.00	available	7	1200
15	Chevrolet	Malibu	2022	47.00	available	4	1800
16	Hyundai	Sonata	2023	49.00	notavailable	7	1400
17	BMW	3 Series	2023	60.00	available	7	2499
18	Mercedes	C-Class	2022	58.00	available	8	2599
19	Audi	A4	2022	55.00	notavailable	4	2500
20	Lexus	ES	2023	54.00	available	4	2500
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Creating Customer Table -

```
1 • create table Customer (
2     customerID int primary key not null auto_increment,
3     firstName varchar(255),
4     lastName varchar(255),
5     email varchar(255),
6     phoneNumber varchar(20)
7 );
8
9 • desc Customer;
```

Field	Type	Null	Key	Default	Extra
customerID	int	NO	PRI	NULL	auto_increment
firstName	varchar(255)	YES		NULL	
lastName	varchar(255)	YES		NULL	
email	varchar(255)	YES		NULL	
phoneNumber	varchar(20)	YES		NULL	

Limit to 1000 rows

1

•

insert into customer (firstName, lastName, email, phoneNumber)

2

values

3

('John', 'Doe', 'johndoe@example.com', '+91-9876543210'),

4

('Jane', 'Smith', 'janesmith@example.com', '+91-8765432109'),

5

('Robert', 'Johnson', 'robert@example.com', '+91-7654321098'),

6

('Sarah', 'Brown', 'sarah@example.com', '+91-6543210987'),

7

('David', 'Lee', 'david@example.com', '+91-9876543210'),

8

('Laura', 'Hall', 'laura@example.com', '+91-8765432109'),

9

('Michael', 'Davis', 'michael@example.com', '+91-7654321098'),

10

('Emma', 'Wilson', 'emma@example.com', '+91-6543210987'),

11

('William', 'Taylor', 'william@example.com', '+91-9876543210'),

12

('Olivia', 'Adams', 'olivia@example.com', '+91-8765432109');

13

Result Grid

Filter Rows:

Edit

Export/Import

Wrap Cell Content

	customerID	firstName	lastName	email	phoneNumber
▶	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
	11	John	Doe	johndoe@example.com	+91-987654...

Result Grid

Form Editor

Field Types

Creating Lease Table -

Limit to 1000 rows

1

•

create table Lease (

2

leaseID int primary key not null auto_increment,

3

vehicleID int,

4

customerID int,

5

startDate date,

6

endDate date,

7

type enum('dailylease', 'monthlylease'),

8

foreign key (vehicleID) references Vehicle(vehicleID),

9

foreign key (customerID) references Customer(customerID)

10

);

11

12

•

desc Lease;

Result Grid

Filter Rows:

Export

Wrap Cell Content

Field	Type	Null	Key	Default	Extra
leaseID	int	NO	PRI	NULL	auto_increment
vehicleID	int	YES	MUL	NULL	
customerID	int	YES	MUL	NULL	
startDate	date	YES		NULL	
endDate	date	YES		NULL	
type	enum('dailylease','monthlylease')	YES		NULL	

Result Grid

Form Editor

Field Types

Limit to 1000 rows

1

•

insert into lease (vehicleID, customerID, startDate, endDate, type)

2

values

3

(14, 4, '2023-04-20', '2023-04-30', 'Monthly'),

4

(15, 5, '2023-05-05', '2023-05-10', 'Daily'),

5

(16, 3, '2023-06-15', '2023-06-30', 'Monthly'),

6

(17, 7, '2023-07-01', '2023-07-10', 'Daily'),

7

(18, 8, '2023-08-12', '2023-08-15', 'Monthly'),

8

(19, 3, '2023-09-07', '2023-09-10', 'Daily'),

9

(20, 10, '2023-10-10', '2023-10-31', 'Monthly');

10

11

•

select * from lease;

Result Grid

Filter Rows:

Edit

Export/Import

Wrap Cell Content

	leaseID	vehicleID	customerID	startDate	endDate	type
▶	1	11	1	2023-01-01	2023-01-05	Daily
	2	12	2	2023-02-15	2023-02-28	Monthly
	3	13	3	2023-03-10	2023-03-15	Daily
	4	14	4	2023-04-20	2023-04-30	Monthly
	5	15	5	2023-05-05	2023-05-10	Daily
	6	16	3	2023-06-15	2023-06-30	Monthly
	7	17	7	2023-07-01	2023-07-10	Daily
	8	18	8	2023-08-12	2023-08-15	Monthly
	9	19	3	2023-09-07	2023-09-10	Daily
	10	20	10	2023-10-10	2023-10-31	Monthly

Result Grid

Form Editor

Field Types

Creating Payment Table -

The screenshot shows the SQL Server Enterprise Manager interface. The top toolbar includes icons for file operations, search, and execution. The main window displays a SQL script for creating a table named 'Payment'.

```

1 • create table Payment (
2     paymentID int primary key not null auto_increment,
3     leaseID int,
4     paymentDate date,
5     amount decimal(10, 2),
6     foreign key (leaseID) references Lease(leaseID)
7 );
8
9 • desc Payment;
  
```

Below the script, the 'Result Grid' is visible, showing the table structure for 'Payment'.

Field	Type	Null	Key	Default	Extra
paymentID	int	NO	PRI	NULL	auto_increment
leaseID	int	YES	MUL	NULL	
paymentDate	date	YES		NULL	
amount	decimal(10,2)	YES		NULL	

The screenshot shows a SQL Studio interface. The top toolbar includes icons for file operations, search, and execution. Below the toolbar, a query editor displays the following SQL code:

```
1 • insert into payment (leaseID, paymentDate, amount)
2 values
3 (1, '2023-01-03', 200.00),
4 (2, '2023-02-20', 1000.00),
5 (3, '2023-03-12', 75.00),
6 (4, '2023-04-25', 900.00),
7 (5, '2023-05-07', 60.00),
8 (6, '2023-06-18', 1200.00),
9 (7, '2023-07-03', 40.00),
10 (8, '2023-08-14', 1100.00),
11 (9, '2023-09-09', 80.00),
12 (10, '2023-10-25', 1500.00);
13
14 • select * from payment;
```

Below the query editor, the "Result Grid" tab is active, displaying the results of the SELECT query in a table format:

	paymentID	leaseID	paymentDate	amount
▶	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
*	NULL	NULL	NULL	NULL

The bottom toolbar contains options for editing, exporting/importing, wrapping cell content, and switching between Result Grid, Form Editor, and Field Types views.

Tasks

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

1 • update vehicle

2 • set dailyRate = 68.00

3 • where make = 'Mercedes';

4

5 • select * from vehicle;

Result Grid

vehicleID	make	model	year	dailyRate	status	passengerCapacity	engineCapacity
11	Toyota	Camry	2022	50.00	available	4	1450
12	Honda	Civic	2023	45.00	available	7	1500
13	Ford	Focus	2022	48.00	notavailable	4	1400
14	Nissan	Altima	2023	52.00	available	7	1200
15	Chevrolet	Malibu	2022	47.00	available	4	1800
16	Hyundai	Sonata	2023	49.00	notavailable	7	1400
17	BMW	3 Series	2023	60.00	available	7	2499
18	Mercedes	C-Class	2022	68.00	available	8	2599

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

1 • SET FOREIGN_KEY_CHECKS = 0;

2

3 • DELETE FROM payment

4 • WHERE leaseID IN (SELECT leaseID FROM lease WHERE customerID = 3);

5

6 • DELETE FROM lease WHERE customerID = 3;

7

8 • DELETE FROM customer WHERE customerID = 3;

9

10 • SET FOREIGN_KEY_CHECKS = 1;

Result Grid

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

11

12 • select * from lease;

Result Grid

leaseID	vehicleID	customerID	startDate	endDate	type
1	11	1	2023-01-01	2023-01-05	Daily
2	12	2	2023-02-15	2023-02-28	Monthly
4	14	4	2023-04-20	2023-04-30	Monthly
5	15	5	2023-05-05	2023-05-10	Daily
7	17	7	2023-07-01	2023-07-10	Daily
8	18	8	2023-08-12	2023-08-15	Monthly

11

12 • select * from payment;

Result Grid

paymentID	leaseID	transactionDate	amount
1	1	2023-01-03	200.00
4	4	2023-04-25	900.00
5	5	2023-05-07	60.00
7	7	2023-07-03	40.00
8	8	2023-08-14	1100.00
10	10	2023-10-25	1500.00

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

Limit to 1000 rows

```

1 • ALTER TABLE Payment
2   CHANGE COLUMN paymentDate transactionDate DATE;
3
4 • desc payment;
```

Field	Type	Null	Key	Default	Extra
paymentID	int	NO	PRI	NULL	auto_increment
leaseID	int	YES	MUL	NULL	
transactionDate	date	YES		NULL	
amount	decimal(10,2)	YES		NULL	

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

Result Grid Form

4. Find a specific customer by email.

```
1 • SELECT *
2 FROM Customer
3 WHERE email = 'robert@example.com';
```

5. Get active leases for a specific customer.

```
1 SELECT
2     lease.leaseID, vehicle.vehicleID, lease.customerID
3 FROM
4     lease
5 JOIN
6     vehicle ON lease.vehicleID = vehicle.vehicleID
7 WHERE
8     lease.customerID = 4
9     AND vehicle.status = 'available'
10    AND CURDATE() BETWEEN lease.startDate AND lease.endDate;
```

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

The screenshot shows the SQL Server Enterprise Manager interface. At the top, there's a toolbar with various icons. Below it, a query window displays the following SQL code:

```
1 • SELECT Payment.*  
2 FROM Payment  
3 INNER JOIN Lease ON Payment.leaseID = Lease.leaseID  
4 INNER JOIN Customer ON Lease.customerID = Customer.customerID  
5 WHERE Customer.phoneNumber = '555-432-1098';
```

Below the query window, there's a "Result Grid" section. It includes a "Filter Rows:" input field, an "Export:" button, and a "Wrap Cell Content:" checkbox. The result grid itself contains one row of data:

	paymentID	leaseID	transactionDate	amount
▶	8	8	2023-08-14	1100.00

In the bottom right corner, there's a blue button labeled "Result Grid" with a grid icon.

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

The screenshot shows a database management interface. At the top, a toolbar contains icons for file operations, search, and execution. Below the toolbar, the SQL editor displays the following query:

```
1 • SELECT AVG(dailyRate) AS averageDailyRate
2 FROM vehicle
3 WHERE status = 'available';
```

Below the editor, a horizontal scrollbar is visible. At the bottom, the 'Result Grid' tab is active, showing a table with one column, 'averageDailyRate', and one row with the value '53.714286'. The bottom status bar includes a 'Filter Rows' input field, an 'Export' button, and a 'Wrap Cell Content' checkbox.

8. Find the car with the highest daily rate.

```

1 • SELECT vehicle.*
2   FROM vehicle
3  WHERE dailyRate = (
4         select MAX(dailyRate) from vehicle
5     );

```

vehicleID	make	model	year	dailyRate	status	passengerCapacity	engineCapacity
18	Mercedes	C-Class	2022	68.00	available	8	2599

9. Retrieve all cars leased by a specific customer.

```
1 select c.customerID, c.firstName, c.lastName, v.*
2 from vehicle v
3 join lease l on l.vehicleID = v.vehicleID
4 join customer c on c.customerID = l.customerID
5 where c.customerID = 4;
```

Result Grid

	customerID	firstName	lastName	vehicleID	make	model	year	dailyRate	status	passengerCapacity	engineCapacity
▶	4	Sarah	Brown	14	Nissan	Altima	2023	52.00	available	7	1200

10. Find the details of the most recent lease.

```

1 SELECT *
2 FROM lease
3 WHERE startDate = (
4     SELECT MAX(startDate)
5     FROM lease
6 )
7

```

leaseID	vehicleID	customerID	startDate	endDate	type
10	20	10	2023-10-10	2023-10-31	Monthly

11. List all payments made in the year 2023.

Limit to 1000 rows

1

select *

2

from payment

3

where year(transactionDate) = 2023;

Result Grid

Filter Rows:

Edit:

Export/Import:

Wrap Cell Content:

	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
	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
*	NULL	NULL	NULL	NULL

Result Grid

Form Editor

Field Types

12. Retrieve customers who have not made any payments.

Limit to 1000 rows

1

SELECT * FROM Customer

2

WHERE customerID NOT IN (SELECT DISTINCT customerID FROM Payment);

3

Result Grid

Filter Rows:

Edit:

Export/Import:

Wrap Cell Content:

	customerID	firstName	lastName	email	phoneNumber
*	NULL	NULL	NULL	NULL	NULL

Result Grid

Form Editor

Field Types

13. Retrieve Car Details and Their Total Payments.

Limit to 1000 rows

1

select v.*, coalesce(sum(p.amount), 0) as total_payment

2

from vehicle v

3

left join lease l on v.vehicleID = l.vehicleID

4

left join payment p on p.leaseID = l.leaseID

5

group by v.vehicleID;

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

	vehicleID	make	model	year	dailyRate	status	passengerCapacity	engineCapacity	total_payment
▶	11	Toyota	Camry	2022	50.00	available	4	1450	200.00
	12	Honda	Civic	2023	45.00	available	7	1500	0.00
	13	Ford	Focus	2022	48.00	notavailable	4	1400	75.00
	14	Nissan	Altima	2023	52.00	available	7	1200	900.00
	15	Chevrolet	Malibu	2022	47.00	available	4	1800	60.00
	16	Hyundai	Sonata	2023	49.00	notavailable	7	1400	1200.00
	17	BMW	3 Series	2023	60.00	available	7	2499	40.00
	18	Mercedes	C-Class	2022	68.00	available	8	2599	1100.00
	19	Audi	A4	2022	55.00	notavailable	4	2500	80.00
	20	Lexus	ES	2023	54.00	available	4	2500	1500.00

Result Grid

Form Editor

Field Types

14. Calculate Total Payments for Each Customer.

Limit to 1000 rows

1

select c.*, coalesce(SUM(p.amount), 0) as totalPayment

2

from customer c

3

left join lease l on c.customerID = l.customerID

4

left join payment p on p.leaseID = l.leaseID

5

group by c.customerID;

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

	customerID	firstName	lastName	email	phoneNumber	totalPayment
▶	1	John	Doe	john.doe@example.com	555-555-5555	200.00
	2	Jane	Smith	jane.smith@example.com	555-123-4567	0.00
	3	Robert	Johnson	robert@example.com	555-789-1234	1355.00
	4	Sarah	Brown	sarah@example.com	555-456-7890	900.00
	5	David	Lee	david@example.com	555-987-6543	60.00
	6	Laura	Hall	laura@example.com	555-234-5678	0.00
	7	Michael	Davis	michael@example.com	555-876-5432	40.00
	8	Emma	Wilson	emma@example.com	555-432-1098	1100.00
	9	William	Taylor	william@example.com	555-321-6547	0.00
	10	Olivia	Adams	olivia@example.com	555-765-4321	1500.00

Result Grid

Form Editor

Field Types

15. List Car Details for Each Lease.

Limit to 1000 rows

1

select l.leaseID, v.*

2

from lease l

3

left join vehicle v on v.vehicleID = l.vehicleID;

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

	leaseID	vehicleID	make	model	year	dailyRate	status	passengerCapacity	engineCapacity
▶	1	11	Toyota	Camry	2022	50.00	available	4	1450
	2	12	Honda	Civic	2023	45.00	available	7	1500
	3	13	Ford	Focus	2022	48.00	notavailable	4	1400
	4	14	Nissan	Altima	2023	52.00	available	7	1200
	5	15	Chevrolet	Malibu	2022	47.00	available	4	1800
	6	16	Hyundai	Sonata	2023	49.00	notavailable	7	1400
	7	17	BMW	3 Series	2023	60.00	available	7	2499
	8	18	Mercedes	C-Class	2022	68.00	available	8	2599
	9	19	Audi	A4	2022	55.00	notavailable	4	2500
	10	20	Lexus	ES	2023	54.00	available	4	2500

Result Grid

Form Editor

Field Types

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

Limit to 1000 rows

1

select l.leaseID, c.firstName, v.make

2

from lease l

3

left join vehicle v on v.vehicleID = l.vehicleID

4

left join customer c on c.customerID = l.customerID

5

where l.endDate >= curdate();

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

	leaseID	firstName	make
--	---------	-----------	------

Result Grid

Form Editor

Field Types

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

```
1 • SELECT c.customerID, c.firstName, c.lastName, COALESCE(SUM(p.amount), 0) AS totalPayments
2 FROM Customer c
3 LEFT JOIN Lease l ON c.customerID = l.customerID
4 LEFT JOIN Payment p ON l.leaseID = p.leaseID
5 GROUP BY c.customerID
6 ORDER BY totalPayments DESC
7 LIMIT 1;
```

Result Grid

	customerID	firstName	lastName	totalPayments
▶	10	Olivia	Adams	1500.00

18. List All Cars with Their Current Lease Information.

Limit to 1000 rows

```

1 • select v.vehicleID, l.*
2   from vehicle v
3  left join lease l on l.vehicleID = v.vehicleID;

```

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

	vehicleID	leaseID	vehicleID	customerID	startDate	endDate	type
▶	11	1	11	1	2023-01-01	2023-01-05	Daily
	12	2	12	2	2023-02-15	2023-02-28	Monthly
	13	3	13	3	2023-03-10	2023-03-15	Daily
	14	4	14	4	2023-04-20	2023-04-30	Monthly
	15	5	15	5	2023-05-05	2023-05-10	Daily
	16	6	16	3	2023-06-15	2023-06-30	Monthly
	17	7	17	7	2023-07-01	2023-07-10	Daily
	18	8	18	8	2023-08-12	2023-08-15	Monthly
	19	9	19	3	2023-09-07	2023-09-10	Daily
	20	10	20	10	2023-10-10	2023-10-31	Monthly

Form Editor
Field Types