

Coding Challenge Output

Car Rental System

1. Database Creation

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

- Schemas:** Local instance MySQL80 (car...), Schemas: sakila, siedb, sys, virtual_art_gallery, world.
- Query Editor:** Contains the SQL code for creating the database and its schema. The code includes:
 - Database Creation
 - create database Car_Rental_System;
 - use car_rental_system;
 - Database Schema Creation
 - create table Vehicle(VehicleId int primary key, Make varchar(50), Model varchar(50), Year int, Daily_Rate decimal(10,2), Status enum('available','unavailable'), Pass_Capacity int, Eng_Capacity int);
 - desc Vehicle;
 - create table Customer(CustomerId int primary key, F_name varchar(50), L_name varchar(50), email varchar(50) unique, ph_num varchar(50) unique);
 - desc Customer;
 - create table Lease(LeaseId int primary key, VehicleId int, CustomerId int, St_date date, End_date date, LeaseType enum('Daily', 'Monthly'), foreign key (VehicleId) references Vehicle(VehicleId), foreign key (CustomerId) references Customer(CustomerId));
 - desc Lease;
 - create table Payment(PaymentId int primary key, LeaseId int, Pay_Date date, amt decimal(10,2), foreign key (LeaseId) references Lease(LeaseId));
 - desc Payment;
- Output:** Shows the execution results:
 - Action Output: 95, 96, 97 (Success)
 - Message: Error Code: 1175. You are using safe update mode and you tried to update a table without a WHERE that uses a column that is foreign key (from another table). You can turn off this error by setting sql_mode='NO_AUTO_VALUE_ON_ZERO'.
 - Duration / Fetch: 0.000 sec, 0.234 sec, 0.015 sec

2. Schema Creation

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

- Schemas:** Local instance MySQL80 (car...), Schemas: sakila, siedb, sys, virtual_art_gallery, world.
- Query Editor:** Contains the SQL code for creating the database and its schema. The code includes:
 - Database Creation
 - create database Car_Rental_System;
 - use car_rental_system;
 - Database Schema Creation
 - create table Vehicle(VehicleId int primary key, Make varchar(50), Model varchar(50), Year int, Daily_Rate decimal(10,2), Status enum('available','unavailable'), Pass_Capacity int, Eng_Capacity int);
 - desc Vehicle;
- Result Grid:** Displays the schema of the Vehicle table.

Field	Type	Null	Key	Default	Extra
VehicleId	int	NO	PRI	NULL	
Make	varchar(50)	YES	NULL		
Model	varchar(50)	YES	NULL		
Year	int	YES	NULL		
Daily_Rate	decimal(10,2)	YES	NULL		
Status	enum('available','unavailable')	YES	NULL		
Pass_Capacity	int	YES	NULL		
Eng_Capacity	int	YES	NULL		
- Result 39:** Shows the execution results:
 - Action Output: 100, 101, 102 (Success)
 - Message: Error Code: 1007. Can't create database 'car_rental_system'; database exists
 - Duration / Fetch: 0.000 sec, 0.062 sec, 0.016 sec / 0.000 sec

File Edit View Query Database Server Tools Scripting Help

Navigator Schemas Administration Schemas Information No object selected Object Info Session Query Completed

Class Work Daily Task Case Study Car_Rental_System

```

12 • create table Customer(CustomerId int primary key, F_name varchar(50), L_name varchar(50), email varchar(50) unique, ph_num varchar(50) unique);
13
14 • desc Customer;
15
16 • create table Lease(LeaseId int primary key, VehicleId int, CustomerId int, St_date date, End_date date, LeaseType enum('Daily', 'Monthly'), foreign key (VehicleId) references Vehicle(VehicleId), foreign key (CustomerId) references Customer(CustomerId));
17
18
19 • desc Lease;
20
21 • create table Payment(PaymentId int primary key, LeaseId int, Pay_Date date, amt decimal(10,2),

```

Result Grid | Filter Rows: Export: Wrap Cell Content: 15

Field	Type	Null	Key	Default	Extra
LeaseId	int	NO	PRI	NULL	
VehicleId	int	YES	MUL	NULL	
CustomerId	int	YES	MUL	NULL	
St_date	date	YES	MUL	NULL	
End_date	date	YES	MUL	NULL	
LeaseType	enum(Daily,Monthly)	YES		NULL	

Result 41 x

Action Output

#	Time	Action	Message	Duration / Fetch
104	13:30:33	desc Customer	5 row(s) returned	0.000 sec / 0.000 sec
105	13:30:53	create table Lease(LeaseId int primary key, VehicleId int, CustomerId int, St_date date, End_date date, LeaseT... 0 row(s) affected		0.109 sec
106	13:30:53	desc Lease	6 row(s) returned	0.000 sec / 0.000 sec

Object Info Session Query Completed

File Edit View Query Database Server Tools Scripting Help

Navigator Schemas Administration Schemas Information No object selected Object Info Session Query Completed

Class Work Daily Task Case Study Car_Rental_System

```

18
19 • desc Lease;
20
21 • create table Payment(PaymentId int primary key, LeaseId int, Pay_Date date, amt decimal(10,2), foreign key (LeasedId) references Lease(LeasedId));
22
23
24 • desc payment;
25
26 -- Insertion of Values into Database
27 • insert into Vehicle values

```

Result Grid | Filter Rows: Export: Wrap Cell Content: 15

Field	Type	Null	Key	Default	Extra
PaymentId	int	NO	PRI	NULL	
LeasedId	int	YES	MUL	NULL	
Pay_Date	date	YES		NULL	
amt	decimal(10,2)	YES		NULL	

Result 42 x

Action Output

#	Time	Action	Message	Duration / Fetch
106	13:30:53	desc Lease	6 row(s) returned	0.000 sec / 0.000 sec
107	13:31:09	create table Payment(PaymentId int primary key, LeasedId int, Pay_Date date, amt decimal(10,2), foreign key (L... 0 row(s) affected		0.062 sec
108	13:31:09	desc payment	4 row(s) returned	0.000 sec / 0.000 sec

Object Info Session Query Completed

3. Insertion of Values

Schemas

```

19 • insert into payment values
20 (1, 1, '2023-01-03', 200.00),
21 (2, 2, '2023-02-28', 1000.00),
22 (3, 3, '2023-03-12', 75.00),
23 (4, 4, '2023-04-25', 900.00),
24 (5, 5, '2023-05-07', 60.00),
25 (6, 6, '2023-06-18', 1200.00),
26 (7, 7, '2023-07-03', 40.00),
27 (8, 8, '2023-08-14', 1100.00),
28 (9, 9, '2023-09-09', 80.00),
29 (10, 10, '2023-10-25', 1500.00);
30
31 • select * from payment;
32

```

Result Grid

PaymentId	LeasedId	Pay_Date	amt
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

payment 46 x

Object Info Session

Action Output

Query Completed

Schemas

```

55 • insert into lease values
56 (1, 1, 1, '2023-01-01', '2023-01-05', 'Daily'),
57 (2, 2, 2, '2023-02-15', '2023-02-28', 'Monthly'),
58 (3, 3, 3, '2023-03-10', '2023-03-15', 'Daily'),
59 (4, 4, 4, '2023-04-20', '2023-04-30', 'Monthly'),
60 (5, 5, 5, '2023-05-05', '2023-05-10', 'Daily'),
61 (6, 6, 6, '2023-06-15', '2023-06-30', 'Monthly'),
62 (7, 7, 7, '2023-07-01', '2023-07-10', 'Daily'),
63 (8, 8, 8, '2023-08-12', '2023-08-15', 'Monthly'),
64 (9, 9, 9, '2023-09-07', '2023-09-10', 'Daily'),
65 (10, 10, 10, '2023-10-10', '2023-10-31', 'Monthly');
66
67 • select * from lease;
68

```

Result Grid

LeasedId	VehicleId	CustomerId	St_date	End_date	LeaseType
1	1	1	2023-01-01	2023-01-05	Daily
2	2	2	2023-02-15	2023-02-28	Monthly
3	3	3	2023-03-10	2023-03-15	Daily
4	4	4	2023-04-20	2023-04-30	Monthly
5	5	5	2023-05-05	2023-05-10	Daily
6	4	3	2023-06-15	2023-06-30	Monthly
7	7	7	2023-07-01	2023-07-10	Daily
8	8	8	2023-08-12	2023-08-15	Monthly
9	3	3	2023-09-07	2023-09-10	Daily
10	10	10	2023-10-10	2023-10-31	Monthly

lease 45 x

Object Info Session

Action Output

Query Completed

Schemas

```

41 • insert into Customer values
42 (1, 'John', 'Doe', 'johndoe@example.com', '555-555-5555'),
43 (2, 'Jane', 'Smith', 'janessmith@example.com', '555-123-4567'),
44 (3, 'Robert', 'Johnson', 'robert@example.com', '555-789-1234'),
45 (4, 'Sarah', 'Brown', 'sarah@example.com', '555-456-7890'),
46 (5, 'David', 'Lee', 'davide@example.com', '555-987-6543'),
47 (6, 'Laura', 'Hall', 'laura@example.com', '555-234-5678'),
48 (7, 'Michael', 'Davis', 'michael@example.com', '555-876-5432'),
49 (8, 'Emma', 'Wilson', 'emma@example.com', '555-432-1098'),
50 (9, 'William', 'Taylor', 'william@example.com', '555-321-6547'),
51 (10, 'Olivia', 'Adams', 'olivia@example.com', '555-765-4321');
52
53 • select * from Customer;
54

```

Result Grid

CustomerId	F_name	L_name	email	ph_num
1	John	Doe	johndoe@example.com	555-555-5555
2	Jane	Smith	janessmith@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	davide@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

Customer 44 x

Object Info Session

Action Output

Task 1 - Updation of daily rate for a Mercedes car to 68.

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

- Navigator:** Shows the schema `car_rental_system` containing tables, views, stored procedures, and functions.
- Class Work:** The current tab is `Case_Study`.
- Code Editor:** Displays a SQL script with several numbered comments and statements. The script includes:
 - Updating vehicle daily rate to 68.
 - Deleting a specific customer and associated leases and payments.
 - Renaming the "Pay_Date" column to "transactionDate".
 - Finding a specific customer by email.
- Output:** Shows the execution log with the following entries:

#	Time	Action	Message	Duration / Fetch
113	13:32:54	select * from Customer LIMIT 0, 1000	10 row(s) returned	0.00 sec / 0.000 sec
114	13:33:27	insert into lease values (1, 1, 1, '2023-01-01', '2023-01-05', 'Daily'), (2, 2, 2, '2023-02-15', '2023-02-28', 'Monthly')	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.016 sec
115	13:33:27	select * from lease LIMIT 0, 1000	10 row(s) returned	0.000 sec / 0.000 sec
116	13:33:49	insert into payment values (1, 1, 2023-01-03, 200.00), (2, 2, 2023-02-20, 1000.00), (3, 3, 2023-03-12, 75.00)	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.016 sec
117	13:33:49	select * from payment LIMIT 0, 1000	10 row(s) returned	0.000 sec / 0.000 sec
118	13:34:09	update Vehicle set Daily_Rate=68 where VehicleId=(Select VehicleId from Vehicle where Make='Mercedes' Limit 1)as Temp;	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.015 sec
119	13:34:12	update Vehicle set Daily_Rate=68 where VehicleId=(Select VehicleId from Vehicle where Make='Mercedes' Limit 1)as Temp;	0 row(s) affected Rows matched: 1 Changed: 0 Warnings: 0	0.000 sec

Task 2 - Deletion of a specific customer and all associated leases and payments.

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

- Navigator:** Shows the schema `car_rental_system` containing tables, views, stored procedures, and functions.
- Class Work:** The current tab is `Case_Study`.
- Code Editor:** Displays a SQL script with several numbered comments and statements. The script includes:
 - Updating vehicle daily rate to 68.
 - Deleting a specific customer and associated leases and payments.
 - Renaming the "Pay_Date" column to "transactionDate".
 - Finding a specific customer by email.
- Output:** Shows the execution log with the following entries:

#	Time	Action	Message	Duration / Fetch
116	13:33:49	insert into payment values (1, 1, 2023-01-03, 200.00), (2, 2, 2023-02-20, 1000.00), (3, 3, 2023-03-12, 75.00)	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.016 sec
117	13:33:49	select * from payment LIMIT 0, 1000	10 row(s) returned	0.000 sec / 0.000 sec
118	13:34:09	update Vehicle set Daily_Rate=68 where VehicleId=(Select VehicleId from Vehicle where Make='Mercedes' Limit 1)as Temp;	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.015 sec
119	13:34:12	update Vehicle set Daily_Rate=68 where VehicleId=(Select VehicleId from Vehicle where Make='Mercedes' Limit 1)as Temp;	0 row(s) affected Rows matched: 1 Changed: 0 Warnings: 0	0.000 sec
120	13:34:42	delete from payment where LeaseId=(select LeaseId from Lease where CustomerId=1)	1 row(s) affected	0.016 sec
121	13:34:42	delete from Lease where CustomerId=1	1 row(s) affected	0.000 sec
122	13:34:42	delete from customer where CustomerId=1	1 row(s) affected	0.000 sec

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

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

- File Edit View Query Database Server Tools Scripting Help**
- Navigator** pane: Schemas (car_rental_system), Tables (sakila, stdb, sys, virtual_art_gallery, world).
- Class Work Daily Task Case_Study Car_Rental_System**
- Text Editor** pane (Query tab):

```

87 -- 2. Deletion of a specific customer and all associated leases and payments.
88 • delete from payment where LeasedId=(select LeasedId from Lease where CustomerId=1);
89 • delete from Lease where CustomerId=1;
90 • delete from customer where CustomerId=1;
91
92 -- 3. Rename the "Pay_Date" column in the Payment table to "transactionDate".
93 • alter table payment rename column Pay_Date to transactionDate;
94 • desc payment;
95
96 -- 4. Find a specific customer by email.
97 • select * from customer where email='david@example.com';
98
99 -- 5. Get active leases for a specific customer.

```
- Result Grid** pane: Shows the structure of the Payment table with columns: PaymentId (int, NO, PRI), LeasedId (int, YES, MUL), transactionDate (date, YES), and amt (decimal(10,2), YES).
- Action Output** pane: Shows the execution log with four entries corresponding to the numbered steps in the script.
- Object Info Session**
- Query Completed**

Task 4 - Find a specific customer by email.

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

- File Edit View Query Database Server Tools Scripting Help**
- Navigator** pane: Schemas (car_rental_system), Tables (sakila, stdb, sys, virtual_art_gallery, world).
- Class Work Daily Task Case_Study Car_Rental_System**
- Text Editor** pane (Query tab):

```

87 -- 2. Deletion of a specific customer and all associated leases and payments.
88 • delete from payment where LeasedId=(select LeasedId from Lease where CustomerId=1);
89 • delete from Lease where CustomerId=1;
90 • delete from customer where CustomerId=1;
91
92 -- 3. Rename the "Pay_Date" column in the Payment table to "transactionDate".
93 • alter table payment rename column Pay_Date to transactionDate;
94 • desc payment;
95
96 -- 4. Find a specific customer by email.
97 • select * from customer where email='david@example.com';
98
99 -- 5. Get active leases for a specific customer.

```
- Result Grid** pane: Shows the result of the query `select * from customer where email='david@example.com';` with one row:

CustomerId	F_name	L_name	email	ph_num
5	David	Lee	david@example.com	555-987-6543
- Action Output** pane: Shows the execution log with five entries corresponding to the numbered steps in the script.
- Object Info Session**
- Query Completed**

Task 5 - Get active leases for a specific customer.

```

File Edit View Query Database Server Tools Scripting Help
Navigator Schemas Administration Information No object selected Object Info Session Query Completed
Class Work Daily Task Case_Study Car_Rental_System * Limit to 1000 rows
93 • alter table payment rename column Pay_Date to transactionDate;
94 • desc payment;
95
96 -- 4. Find a specific customer by email.
97 • select * from customer where email='david@example.com';
98
99 -- 5. Get active leases for a specific customer.
100 • select * from lease where CustomerId=7 and End_Date >= '2023-05-10';
101 • select * from lease where CustomerId=7 and End_Date >= curdate();
102
103 -- 6. All payments made by a customer with a specific phone number.
104 • select p.* from payment p join lease l on p.leaseId=l.leaseId join customer c on l.CustomerId=c.CustomerId where c.ph_num='555-987-6543';
105 • select * from payment where LeasedId=(Select LeaseId from Lease where CustomerId=(Select CustomerId from Customer where ph_num='555-432-1098')));
106
107
108
109
110
111

```

Result Grid

LeasedId	VehicleId	CustomerId	St_date	End_date	LeaseType
7	7	7	2023-07-01	2023-07-10	Daily

lease51

Action Output

Time	Action	Message	Duration / Fetch
125	13:35:29 select * from customer where email='david@example.com' LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
126	13:35:46 select * from lease where CustomerId=4 and End_Date >= '2023-05-10' LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec
127	13:36:09 select * from lease LIMIT 0, 1000	9 row(s) returned	0.000 sec / 0.000 sec
128	13:36:58 select * from lease where CustomerId=7 and End_Date >= '2023-05-10' LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

Task 6 - All payments made by a customer with a specific phone number.

```

File Edit View Query Database Server Tools Scripting Help
Navigator Schemas Administration Information No object selected Object Info Session Query Completed
Class Work Daily Task Case_Study Car_Rental_System * Limit to 1000 rows
99 -- 5. Get active leases for a specific customer.
100 • select * from lease where CustomerId=7 and End_Date >= '2023-05-10';
101 • select * from lease where CustomerId=7 and End_Date >= curdate();
102
103 -- 6. All payments made by a customer with a specific phone number.
104 • select p.* from payment p join lease l on p.leaseId=l.leaseId join customer c on l.CustomerId=c.CustomerId where c.ph_num='555-987-6543';
105 • select * from payment where LeasedId=(Select LeaseId from Lease where CustomerId=(Select CustomerId from Customer where ph_num='555-432-1098')));
106
107 -- 7. Calculate the average daily_rate of all available cars.
108 • select avg(Daily_Rate) as Average_Rate from Vehicle;
109
110 -- 8. Vehicle with the highest daily rate.
111 • Select * from Vehicle order by Daily_Rate desc limit 1;
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129

```

Result Grid

PaymentId	LeasedId	transactionDate	amt
8	8	2023-08-14	1100.00

payment 52

Action Output

Time	Action	Message	Duration / Fetch
126	13:35:46 select * from lease where CustomerId=4 and End_Date >= '2023-05-10' LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec
127	13:36:09 select * from lease LIMIT 0, 1000	9 row(s) returned	0.000 sec / 0.000 sec
128	13:36:58 select * from lease where CustomerId=7 and End_Date >= '2023-05-10' LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
129	13:37:24 select * from payment where LeasedId=(Select LeaseId from Lease where CustomerId=(Select CustomerId from Customer where ph_num='555-432-1098'))	1 row(s) returned	0.015 sec / 0.000 sec

Task 7 - Calculate the average daily_rate of all available cars.

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

- Navigator:** Shows the schema `car_rental_system` containing tables like `Lease`, `Customer`, and `Payment`.
- Query Editor:** Contains the following SQL code:

```

182
183 -- 6. All payments made by a customer with a specific phone number.
184 • select p.* from payment p join lease l on p.LeasedId=l.LeasedId join customer c on l.CustomerId=c.CustomerId where c.ph_num='555-987-6543';
185 • select * from payment where LeasedId=(Select LeasedId from Lease where CustomerId=(Select CustomerId from Customer where ph_num='555-432-1098')));
186
187 -- 7. Calculate the average daily_rate of all available cars.
188 • select avg(Daily_Rate) as Average_Rate from Vehicle;
189
190 -- 8. Vehicle with the highest daily rate.
191 • Select * from Vehicle order by Daily_Rate desc limit 1;
192
193 -- 9. Retrieve all Vehicles leased by a specific customer.
194 • select v.* from Vehicle v join lease l on l.LeasedId=v.vehicleId where l.CustomerId=5;
195
196 -- 10. Details of the most recent lease.
197 • select * from lease order by End_Date desc limit 1;
198
199 -- 11. List of payments made in the year 2023.
200 • select * from payment where year(transactionDate)=2023;

```
- Result Grid:** Displays the result of the `select avg(Daily_Rate) as Average_Rate from Vehicle;` query, showing a single row with `Average_Rate` as `$2.80000`.
- Action Output:** Shows the execution log with 4 rows returned in 0.000 sec / 0.000 sec.

Task 8 - Vehicle with the highest daily rate.

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

- Navigator:** Shows the schema `car_rental_system` containing tables like `Lease`, `Customer`, and `Payment`.
- Query Editor:** Contains the following SQL code:

```

108 • select avg(Daily_Rate) as Average_Rate from Vehicle;
109
110 -- 8. Vehicle with the highest daily rate.
111 • Select * from Vehicle order by Daily_Rate desc limit 1;
112
113 -- 9. Retrieve all Vehicles leased by a specific customer.
114 • select v.* from Vehicle v join lease l on l.LeasedId=v.vehicleId where l.CustomerId=5;
115
116 -- 10. Details of the most recent lease.
117 • select * from lease order by End_Date desc limit 1;
118
119 -- 11. List of payments made in the year 2023.
120 • select * from payment where year(transactionDate)=2023;

```
- Result Grid:** Displays the result of the `Select * from Vehicle order by Daily_Rate desc limit 1;` query, showing a single row for vehicle ID 8 with details: Make 'Mercedes', Model 'C-Class', Year '2022', Daily_Rate '68.00', Status 'available', Pass_Capacity '8', and Eng_Capacity '2599'.
- Action Output:** Shows the execution log with 4 rows returned in 0.000 sec / 0.000 sec.

Task 9 - Retrieve all Vehicles leased by a specific customer.

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

- Navigator:** Shows the schema `car_rental_system` containing tables like `sakila`, `siadb`, `sys`, and `virtual_art_gallery`.
- Query Editor:** Contains the following SQL code:

```

108 • select avg(Daily_Rate) as Average_Rate from Vehicle;
109
110 -- 8. Vehicle with the highest daily rate.
111 • Select * from Vehicle order by Daily_Rate desc limit 1;
112
113 -- 9. Retrieve all Vehicles leased by a specific customer.
114 • select v.* from Vehicle v join lease l on l.leaseId=v.vehicleId where l.CustomerId=5;
115
116 -- 10. Details of the most recent lease.
117 • select * from lease order by End_Date desc limit 1;
118
119 -- 11. List of payments made in the year 2023.
120 • select * from payment where year(transactionDate)=2023;

```
- Result Grid:** Displays the results of the query in the `Vehicle` table:

VehicleId	Make	Model	Year	Daily_Rate	Status	Pass_Capacity	Eng_Capacity
5	Chevrolet	Malibu	2022	47.00	available	4	1800
- Action Output:** Shows the execution log with the following entries:

#	Time	Action	Message	Duration / Fetch
129	13:37:24	select * from payment where LeaseId=(Select LeaseId from Lease where CustomerId=(Select CustomerId from Vehicle where VehicleId=5))	1 row(s) returned	0.015 sec / 0.000 sec
130	13:37:37	select avg(Daily_Rate) as Average_Rate from Vehicle LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
131	13:37:51	Select * from Vehicle order by Daily_Rate desc limit 1	1 row(s) returned	0.000 sec / 0.000 sec
132	13:38:31	select v.* from Vehicle v join lease l on l.leaseId=v.vehicleId where l.CustomerId=5 LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

Task 10 - Details of the most recent lease.

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

- Navigator:** Shows the schema `car_rental_system` containing tables like `sakila`, `siadb`, `sys`, and `virtual_art_gallery`.
- Query Editor:** Contains the following SQL code:

```

114 • select v.* from Vehicle v join lease l on l.leaseId=v.vehicleId where l.CustomerId=5;
115
116 -- 10. Details of the most recent lease.
117 • select * from lease order by End_Date desc limit 1;
118
119 -- 11. List of payments made in the year 2023.
120 • select * from payment where year(transactionDate)=2023;
121
122 -- 12. Retrieve customers who have not made any payments.
123 • select * from customer where CustomerId not in (select l.CustomerId from lease l join payment p where p.leaseId=l.leaseId);
124
125 -- 13. Retrieve Vehicle Details with Their Total_Payments.
126 • select v.*, sum(p.amount) as Total_Payment from Vehicle v join lease l on v.VehicleId=l.VehicleId join payment p on l.leaseId=p.PaymentId group by v.VehicleId;

```
- Result Grid:** Displays the results of the query in the `lease` table:

LeaseId	VehicleId	CustomerId	St_date	End_date	LeaseType
10	10	10	2023-10-10	2023-10-31	Monthly
- Action Output:** Shows the execution log with the following entries:

#	Time	Action	Message	Duration / Fetch
130	13:37:37	select avg(Daily_Rate) as Average_Rate from Vehicle LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
131	13:37:51	Select * from Vehicle order by Daily_Rate desc limit 1	1 row(s) returned	0.000 sec / 0.000 sec
132	13:38:31	select v.* from Vehicle v join lease l on l.leaseId=v.vehicleId where l.CustomerId=5 LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
133	13:38:48	select * from lease order by End_Date desc limit 1	1 row(s) returned	0.000 sec / 0.000 sec

Task 11 - List of payments made in the year 2023.

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

- File Edit View Query Database Server Tools Scripting Help**
- Navigator** pane: Schemas (car_rental_system), Administration, Information, No object selected.
- Class Work Daily Task Case_Study Car_Rental_System***
- SQL Editor** pane (top): Contains the following SQL code:


```

117 • select * from lease order by End_Date desc limit 1;
118
119 -- 11. List of payments made in the year 2023.
120 • select * from payment where year(transactionDate)=2023;
121
122 -- 12. Retrieve customers who have not made any payments.
123 • select * from customer where CustomerId not in (select l.CustomerId from lease l join payment p where p.leaseId=l.leaseId);
124
      
```
- Result Grid** pane (bottom): Shows a table with columns PaymentId, LeasedId, transactionDate, amt. The data is:

PaymentId	LeasedId	transactionDate	amt
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
- Output** pane: Action Output showing the execution history of the queries.

Task 12 – Retrieve customers who have not made any payments.

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

- File Edit View Query Database Server Tools Scripting Help**
- Navigator** pane: Schemas (car_rental_system), Administration, Information, No object selected.
- Class Work Daily Task Case_Study Car_Rental_System***
- SQL Editor** pane (top): Contains the following SQL code:


```

128 • select * from payment where year(transactionDate)=2023;
129
130 -- 12. Retrieve customers who have not made any payments.
131 • select * from customer where CustomerId not in (select l.CustomerId from lease l join payment p where p.leaseId=l.leaseId);
132
133 -- 13. Retrieve Vehicle Details with Their Total Payments.
134 • select v.*, sum(p.amt) as Total_Payment from Vehicle v join lease l on v.VehicleId=l.VehicleId join payment p on l.leaseId=p.PaymentId group by v.VehicleId;
135
      
```
- Result Grid** pane (bottom): Shows a table with columns CustomerId, F_name, L_name, email, ph_num. The data is:

CustomerId	F_name	L_name	email	ph_num
6	Laura	Hall	laura@example.com	555-234-5678
9	William	Taylor	william@example.com	555-321-6547
- Output** pane: Action Output showing the execution history of the queries.

Task 13 - Retrieve Vehicle Details with Their Total_Payments.

```

123 • select * from customer where CustomerId not in (select l.CustomerId from lease l join payment p where p.leaseId=l.leaseId);
124
125 -- 13. Retrieve Vehicle Details with Their Total_Payments.
126 • select v.*, sum(p.amt) as Total_Payment from Vehicle v join lease l on v.VehicleId=l.VehicleId join payment p on l.leaseId=p.PaymentId group by v.VehicleId;
127
128 -- 14 Calculate Total Payments for Each Customer.
129 • select c.CustomerID, c.F_name, c.L_name, SUM(p.amt) AS totalPayments from Customer c join Lease l on c.CustomerId = l.CustomerId join Payment p on l.leaseId = p.leaseId
130 group by c.CustomerId;
131

```

VehicelId	Make	Model	Year	Dely_Rate	Status	Pass_Capacity	Eng_Capacity	Total_Payment
2	Honda	Civic	2023	45.00	available	7	1500	1000.00
3	Ford	Focus	2022	48.00	unavailable	4	1400	155.00
4	Nissan	Altima	2023	52.00	available	7	1200	2100.00
5	Chevrolet	Malibu	2022	47.00	available	4	1800	60.00
7	BMW	3 Series	2023	60.00	available	7	2499	40.00
8	Mercedes	C-Class	2022	68.00	available	8	2599	1100.00
10	Lexus	ES	2023	54.00	available	4	2500	1500.00

Result 59 x

Action Output

#	Time	Action	Message	Duration / Fetch
133	13:38:48	select * from lease order by End_Date desc limit 1	1 row(s) returned	0.000 sec / 0.000 sec
134	13:39:04	select * from payment where year(transactionDate)=2023 LIMIT 0, 1000	9 row(s) returned	0.015 sec / 0.000 sec
135	13:39:25	select * from customer where CustomerId not in (select l.CustomerId from lease l join payment p where p.leaseId=l.leaseId)	2 row(s) returned	0.000 sec / 0.000 sec
136	13:39:39	select v.*, sum(p.amt) as Total_Payment from Vehicle v join lease l on v.VehicleId=l.VehicleId join payment p on l.leaseId=p.PaymentId group by v.VehicleId;	7 row(s) returned	0.000 sec / 0.000 sec

Task 14 – Calculate Total Payments for Each Customer.

```

124
125 -- 13. Retrieve Vehicle Details with Their Total_Payments.
126 • select v.*, sum(p.amt) as Total_Payment from Vehicle v join lease l on v.VehicleId=l.VehicleId join payment p on l.leaseId=p.PaymentId group by v.VehicleId;
127
128 -- 14 Calculate Total Payments for Each Customer.
129 • select c.CustomerID, c.F_name, c.L_name, SUM(p.amt) AS totalPayments from Customer c join Lease l on c.CustomerId = l.CustomerId join Payment p on l.leaseId = p.leaseId
130 group by c.CustomerId;
131

```

CustomerID	F_name	L_name	totalPayments
2	Jane	Smith	1000.00
3	Robert	Johnson	1355.00
4	Sarah	Brown	900.00
5	David	Lee	60.00
7	Michael	Davis	40.00
8	Emma	Wilson	1100.00
10	Olivia	Adam	1500.00

Result 60 x

Action Output

#	Time	Action	Message	Duration / Fetch
134	13:39:04	select * from payment where year(transactionDate)=2023 LIMIT 0, 1000	9 row(s) returned	0.015 sec / 0.000 sec
135	13:39:25	select * from customer where CustomerId not in (select l.CustomerId from lease l join payment p where p.leaseId=l.leaseId)	2 row(s) returned	0.000 sec / 0.000 sec
136	13:39:39	select v.*, sum(p.amt) as Total_Payment from Vehicle v join lease l on v.VehicleId=l.VehicleId join payment p on l.leaseId=p.PaymentId group by v.VehicleId;	7 row(s) returned	0.000 sec / 0.000 sec
137	13:39:51	select c.CustomerID, c.F_name, c.L_name, SUM(p.amt) 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;	7 row(s) returned	0.000 sec / 0.000 sec

Task 15 - List Vehicle Details for Each Lease.

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

- Navigator:** Shows the schema `car_rental_system` selected.
- Query Editor:** Contains the following SQL code:

```

127
128 -- 14. Calculate Total Payments for Each Customer.
129 • select c.CustomerID, c.F_name, c.L_name, SUM(p.amt) AS totalPayments from Customer c join Lease l on c.CustomerID = l.CustomerID join Payment p on l.LeaseId = p.LeaseId
130 group by c.CustomerID;
131
132 -- 15. List Vehicle Details for Each Lease.
133 • select l.leaseId, v.* from lease l join vehicle v on l.VehicleId=v.VehicleId;
134

```
- Result Grid:** Displays the results of the second query, showing vehicle details for each lease:

leaseId	VehicelId	Make	Model	Year	Daily_Rate	Status	Pass_Capacity	Eng_Capacity
2	2	Honda	Civic	2023	45.00	available	7	1500
3	3	Ford	Focus	2022	48.00	unavailable	4	1400
9	3	Ford	Focus	2022	48.00	unavailable	4	1400
4	4	Nissan	Altima	2023	52.00	available	7	1200
6	4	Nissan	Altima	2023	52.00	available	7	1200
5	5	Chevrolet	Malibu	2022	47.00	available	4	1800
7	7	BMW	3 Series	2023	60.00	available	7	2499
8	8	Mercedes	C-Class	2022	68.00	available	8	2599
10	10	Lexus	ES	2023	54.00	available	4	2500
- Action Output:** Shows the execution log with four entries corresponding to the queries in the editor.

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

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

- Navigator:** Shows the schema `car_rental_system` selected.
- Query Editor:** Contains the following SQL code:

```

130 group by c.CustomerID;
131
132 -- 15. List Vehicle Details for Each Lease.
133 • select l.leaseId, v.* from lease l join vehicle v on l.VehicleId=v.VehicleId;
134
135 -- 16. Retrieve Details of Active Leases with Customer and Vehicle Information.
136 • select l.*, c.*, v.* from lease l join customer c on l.customerId=c.customerId join vehicle v on v.vehicleId=l.vehicleId where end_date>='2023-04-30';
137

```
- Result Grid:** Displays the results of the third query, showing active lease details with associated customer and vehicle information:

leaseId	VehicelId	CustomerID	St_date	End_date	LeaseType	CustomerId	F_name	L_name	email	ph_num	VehicelId	Make	Model	Year	Daily_Rate	Status	Pass_Capacity
4	4	4	2023-04-20	2023-04-30	Monthly	4	Sarah	Brown	sarah@example.com	555-456-7890	4	Nissan	Altima	2023	52.00	available	4
5	5	5	2023-05-01	2023-05-10	Daily	5	David	Lee	david@example.com	555-879-6543	5	Chevrolet	Malibu	2022	47.00	available	4
6	4	3	2023-06-15	2023-06-30	Monthly	3	Robert	Johnson	robert@example.com	555-799-1234	4	Nissan	Altima	2023	52.00	available	7
7	7	7	2023-07-01	2023-07-10	Daily	7	Michael	Davis	michael@example.com	555-876-5432	7	BMW	3 Series	2023	60.00	available	7
8	8	8	2023-08-12	2023-08-15	Monthly	8	Emma	Wilson	emma@example.com	555-432-1098	8	Mercedes	C-Class	2022	68.00	available	8
9	3	3	2023-09-07	2023-09-10	Daily	3	Robert	Johnson	robert@example.com	555-799-1234	3	Ford	Focus	2022	48.00	unavailable	4
10	10	10	2023-10-10	2023-10-31	Monthly	10	Olivia	Adams	olivia@example.com	555-765-4321	10	Lexus	ES	2023	54.00	available	4
- Action Output:** Shows the execution log with four entries corresponding to the queries in the editor.

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

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

- Navigator:** Shows the schema `car_rental_system` selected.
- Query Editor:** Contains the following SQL code:

```

134
135 -- 16. Retrieve Details of Active Leases with Customer and Vehicle Information.
136 • select l.*, c.*, v.* from lease l join customer c on l.customerId=c.customerId join vehicle v on v.vehicleId=l.vehicleId where end_Date>='2023-04-30';
137
138 -- 17. Customer Who Has Spent the Most on Leases.
139 • select c.*,sum(p.amt) as Total_Spent from customer c join lease l on c.customerId=l.customerId join payment p on l.leaseId=p.leaseId group by c.customerId
140 order by Total_Spent desc limit 1;
141

```
- Result Grid:** Displays the result of the last query, showing one row:

CustomerID	F_name	L_name	email	ph_num	Total_Spent
10	Olivia	Adams	olivia@example.com	555-765-4321	1500.00
- Action Output:** Shows the execution log with four entries, all successful (0 rows returned):

#	Time	Action	Message	Duration / Fetch
137	13:39:51	select c.CustomerID, c.F_name, c.L_name, SUM(p.amt) AS totalPayments from Customer c join Lease l on c.C... 9 row(s) returned	Message	0.000 sec / 0.000 sec
138	13:40:09	select l.leaseId, v.* from lease l join vehicle v on l.VehicleId=v.VehicleId LIMIT 0, 1000	9 row(s) returned	0.016 sec / 0.000 sec
139	13:40:26	select l.*, c.*, v.* from lease l join customer c on l.customerId=c.customerId join vehicle v on v.vehicleId=l.vehicleId where end_Date>='2023-04-30'; 7 row(s) returned	Message	0.000 sec / 0.000 sec
140	13:40:44	select c.*sum(p.amt) as Total_Spent from customer c join lease l on c.customerId=l.customerId join payment p on l.leaseId=p.leaseId group by c.customerId order by Total_Spent desc limit 1; 1 row(s) returned	Message	0.000 sec / 0.000 sec

Task 18 - List of All Vehicles and Their Current Lease Information.

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

- Navigator:** Shows the schema `car_rental_system` selected.
- Query Editor:** Contains the following SQL code:

```

138 -- 17. Customer Who Has Spent the Most on Leases.
139 • select c.*,sum(p.amt) as Total_Spent from customer c join lease l on c.customerId=l.customerId join payment p on l.leaseId=p.leaseId group by c.customerId
140 order by Total_Spent desc limit 1;
141
142 -- 18 List of All Vehicles and Their Current Lease Information.
143 • select v.*, l.* From vehicle v left join lease l on v.vehicleId=l.vehicleId where l.End_Date>='2023-05-10';
144

```
- Result Grid:** Displays the result of the last query, showing ten rows of vehicle and lease information:

VehicledID	Make	Model	Year	Daly_Rate	Status	Pass_Capacity	Eng_Capacity	LeaseId	VehicledID	CustomerID	St_date	End_date	LeaseType
5	Chevrolet	Malibu	2022	47.00	available	4	1800	5	5	5	2023-05-05	2023-05-10	Daily
4	Nissan	Altima	2023	52.00	available	7	1200	6	4	3	2023-06-15	2023-06-30	Monthly
7	BMW	3 Series	2023	60.00	available	7	2499	7	7	7	2023-07-01	2023-07-10	Daily
8	Mercedes	C-Class	2022	68.00	available	8	2699	8	8	8	2023-08-12	2023-08-15	Monthly
3	Ford	Focus	2022	48.00	unavailable	4	1400	9	3	3	2023-09-07	2023-09-10	Daily
10	Lexus	ES	2023	54.00	available	4	2500	10	10	10	2023-10-10	2023-10-31	Monthly
- Action Output:** Shows the execution log with five entries, all successful (9 rows returned):

#	Time	Action	Message	Duration / Fetch
138	13:40:09	select l.leaseId, v.* from lease l join vehicle v on l.VehicleId=v.VehicleId LIMIT 0, 1000	9 row(s) returned	0.016 sec / 0.000 sec
139	13:40:26	select l.*, c.*, v.* from lease l join customer c on l.customerId=c.customerId join vehicle v on v.vehicleId=l.vehicleId where end_Date>='2023-04-30'; 7 row(s) returned	Message	0.000 sec / 0.000 sec
140	13:40:44	select c.*sum(p.amt) as Total_Spent from customer c join lease l on c.customerId=l.customerId join payment p on l.leaseId=p.leaseId group by c.customerId order by Total_Spent desc limit 1; 1 row(s) returned	Message	0.000 sec / 0.000 sec
141	13:40:58	select v.*, l.* From vehicle v left join lease l on v.vehicleId=l.vehicleId where l.End_Date>='2023-05-10' LIMIT 0, ... 6 row(s) returned	Message	0.000 sec / 0.000 sec