

Project: Database Systems

CSE 5330-002 – Spring 2023

Deliverable: Project 2 (Part 3)

Prepared by:

Panini Pande 1002064144

Muskan Jain 1002033280

## TABLE OF CONTENTS

1. LOADING DATA INTO TABLES

2. RETRIEVE AND PRINT DATA IN ALL TABLES

3. DATABASE UPDATE TRANSACTIONS

# 1. Loading data into tables

**Task 1:** For loading data into tables, we created a csv file for each of the six tables, and used the MySQL Connector/Python to import data. MySQL Connector enables Python programs to access MySQL databases, using an API that is compliant with the Python

Database API Specification. The source code for the same is listed below:

```
import csv
import mysql.connector

mydb = mysql.connector.connect(host="acadmysqlb001p.uta.edu", user="mxj3280", password="Mjuta2022!",
database="mxj3280")
mycursor = mydb.cursor()
print(mydb)
print(mycursor)

with open('/Users/ASUS/Downloads/dbdata/Owner.csv') as csvfile:
    csv_data = csv.reader(csvfile)
    next(csv_data, None)
    for rows in csv_data:
        a = int(rows[0])
        print(a)
        mycursor.execute("INSERT INTO Owner VALUES (%s)", [a])
csvfile.close()

with open('/Users/ASUS/Downloads/dbdata/Rental_Company.csv') as csvfile:
    csv_data = csv.reader(csvfile)
    next(csv_data, None)
    for rows in csv_data:
        a = int(rows[0])
        b = rows[1]
```

```
print(a,b)

mycursor.execute("INSERT INTO Rental_Company VALUES (%s,%s)", [a,b])
csvfile.close()


with open('/Users/ASUS/Downloads/dbdata/Bank.csv') as csvfile:
    csv_data = csv.reader(csvfile)
    next(csv_data, None)
    for rows in csv_data:
        a = int(rows[0])
        b = rows[1]
        print(a,b)
        mycursor.execute("INSERT INTO Bank VALUES (%s,%s)", [a,b])
    csvfile.close()


with open('/Users/ASUS/Downloads/dbdata/Individual.csv') as csvfile:
    csv_data = csv.reader(csvfile)
    next(csv_data, None)
    for rows in csv_data:
        a = int(rows[0])
        b = rows[1]
        c = rows[2]
        print(a,b,c)
        mycursor.execute("INSERT INTO Individual VALUES (%s,%s,%s)", [a,b,c])
    csvfile.close()


with open('/Users/ASUS/Downloads/dbdata/CAR.csv') as csvfile:
    csv_data = csv.reader(csvfile)
    next(csv_data, None)
    for rows in csv_data:
        a = int(rows[0])
        b = int(rows[1])
        c = rows[2]
        d = rows[3]
        e = rows[4]
        f = rows[5]
        print(a,b,c,d,e,f)
        mycursor.execute("INSERT INTO CAR VALUES (%s,%s,%s,%s,%s,%s)", [a,b,c,d,e,f])
    csvfile.close()
```

```
with open('/Users/ASUS/Downloads/dbdata/CAR_TYPE.csv') as csvfile:
    csv_data = csv.reader(csvfile)
    next(csv_data, None)
    for rows in csv_data:
        a = int(rows[0])
        b = int(rows[1])
        c = int(rows[2])
        print(a,b,c)
        mycursor.execute("INSERT INTO CAR_TYPE VALUES (%s,%s,%s)", [a,b,c])
csvfile.close()
```

```
with open('/Users/ASUS/Downloads/dbdata/Van.csv') as csvfile:
    csv_data = csv.reader(csvfile)
    next(csv_data, None)
    for rows in csv_data:
        a = int(rows[0])
        print(a)
        mycursor.execute("INSERT INTO Van VALUES (%s)", [a])
csvfile.close()
```

```
with open('/Users/ASUS/Downloads/dbdata/Compact.csv') as csvfile:
    csv_data = csv.reader(csvfile)
    next(csv_data, None)
    for rows in csv_data:
        a = int(rows[0])
        print(a)
        mycursor.execute("INSERT INTO Compact VALUES (%s)", [a])
csvfile.close()
```

```
with open('/Users/ASUS/Downloads/dbdata/Truck.csv') as csvfile:
    csv_data = csv.reader(csvfile)
    next(csv_data, None)
    for rows in csv_data:
        a = int(rows[0])
        print(a)
        mycursor.execute("INSERT INTO Truck VALUES (%s)", [a])
csvfile.close()
```

```
with open('/Users/ASUS/Downloads/dbdata/SUV.csv') as csvfile:
    csv_data = csv.reader(csvfile)
    next(csv_data, None)
```

```
for rows in csv_data:
    a = int(rows[0])
    print(a)
    mycursor.execute("INSERT INTO SUV VALUES (%s)", [a])
csvfile.close()

with open('/Users/ASUS/Downloads/dbdata/Large.csv') as csvfile:
    csv_data = csv.reader(csvfile)
    next(csv_data, None)
    for rows in csv_data:
        a = int(rows[0])
        print(a)
        mycursor.execute("INSERT INTO Large VALUES (%s)", [a])
csvfile.close()

with open('/Users/ASUS/Downloads/dbdata/Medium.csv') as csvfile:
    csv_data = csv.reader(csvfile)
    next(csv_data, None)
    for rows in csv_data:
        a = int(rows[0])
        print(a)
        mycursor.execute("INSERT INTO Medium VALUES (%s)", [a])
csvfile.close()

with open('/Users/ASUS/Downloads/dbdata/Luxury.csv') as csvfile:
    csv_data = csv.reader(csvfile)
    next(csv_data, None)
    for rows in csv_data:
        a = int(rows[0])
        print(a)
        mycursor.execute("INSERT INTO Luxury VALUES (%s)", [a])
csvfile.close()

with open('/Users/ASUS/Downloads/dbdata/Regular.csv') as csvfile:
    csv_data = csv.reader(csvfile)
    next(csv_data, None)
    for rows in csv_data:
        a = int(rows[0])
        print(a)
        mycursor.execute("INSERT INTO Regular VALUES (%s)", [a])
csvfile.close()
```

```
with open('/Users/ASUS/Downloads/dbdata/RENTS.csv') as csvfile:
    csv_data = csv.reader(csvfile)
    next(csv_data, None)
    for rows in csv_data:
        a = int(rows[0])
        b = int(rows[1])
        c = rows[2]
        d = rows[3]
        e = int(rows[4])
        f = int(rows[5])
        g = int(rows[6])
        h = int(rows[7])
        i = int(rows[8])
        j = int(rows[9])
        print(a,b,c,d,e,f,g,h,i,j)
        mycursor.execute("INSERT INTO RENTS VALUES (%s,%s,%s,%s,%s,%s,%s,%s,%s,%s)", [a,b,c,d,e,f,g,h,i,j])
csvfile.close()
```

```
with open('/Users/ASUS/Downloads/dbdata/CUSTOMER.csv') as csvfile:
    csv_data = csv.reader(csvfile)
    next(csv_data, None)
    for rows in csv_data:
        a = int(rows[0])
        b = int(rows[1])
        print(a,b)
        mycursor.execute("INSERT INTO CUSTOMER VALUES (%s,%s)", [a,b])
csvfile.close()
```

```
with open('/Users/ASUS/Downloads/dbdata/Individual_Cust.csv') as csvfile:
    csv_data = csv.reader(csvfile)
    next(csv_data, None)
    for rows in csv_data:
        a = int(rows[0])
        b = rows[1]
        c = rows[2]
        print(a,b,c)
        mycursor.execute("INSERT INTO Individual_Cust VALUES (%s,%s,%s)", [a,b,c])
csvfile.close()
```

```
with open('/Users/ASUS/Downloads/dbdata/Company.csv') as csvfile:
    csv_data = csv.reader(csvfile)
    next(csv_data, None)
```

```

for rows in csv_data:
    a = int(rows[0])
    b = rows[1]
    print(a,b)
    mycursor.execute("INSERT INTO Company VALUES (%s,%s)", [a,b])
csvfile.close()

mydb.commit()

```

## 2. Retrieve and Print data in all tables

**Task 2:** To retrieve and print data from all tables

Initial Customer Table

```

mysql> select * from customer;
+-----+-----+
| Cust_ID | Phone |
+-----+-----+
|      11 | 1234  |
|      22 | 2345  |
|      33 | 3456  |
|      44 | 4567  |
|      55 | 5678  |
|      66 | 6789  |
|      77 | 7890  |
|      88 | 8901  |
|      99 | 9012  |
|     100 | 1023  |
+-----+-----+
10 rows in set (0.01 sec)

```



```
mysql> select * from company;
+-----+-----+
| Cust_ID | CName |
+-----+-----+
|      11 | A     |
|      22 | B     |
|      33 | C     |
|      44 | D     |
|      55 | E     |
|      66 | F     |
|      77 | G     |
|      88 | H     |
|      99 | I     |
|     100 | J     |
+-----+-----+
10 rows in set (0.01 sec)
```

Initial Company Table

```
mysql> select * from car_type
-> ;
+-----+-----+-----+
| Type_ID | Daily_Rate | Weekly_Rate |
+-----+-----+-----+
|    1001 |      100   |      200   |
|    1002 |      110   |      220   |
|    1003 |      120   |      240   |
|    1004 |      130   |      260   |
|    1005 |      140   |      280   |
|    1006 |      150   |      300   |
|    1007 |      160   |      320   |
|    1008 |      170   |      340   |
|    1009 |      180   |      360   |
|    1010 |      190   |      380   |
+-----+-----+-----+
10 rows in set (0.02 sec)
```

Initial Car\_type Table

```
mysql> select * from car;
```

Vehicle_ID	Owner_ID	Avail_StartDate	Avail_EndDate	Model	Year
1001	1	2023-04-01	2023-04-02	Toyota	2014
1002	2	2023-04-02	2023-04-03	Nissan	2015
1003	3	2023-04-03	2023-04-04	Honda	2016
1004	4	2023-04-04	2023-04-05	Ford	2017
1005	5	2023-04-05	2023-04-06	Chevrolet	2018
1006	6	2023-04-06	2023-04-07	Volkswagen	2019
1007	7	2023-04-07	2023-04-08	Hyundai	2020
1008	8	2023-04-08	2023-04-09	Kia	2021
1009	9	2023-04-09	2023-04-10	MercedesBenz	2022
1010	10	2023-04-10	2023-04-11	BMW	2023

```
10 rows in set (0.02 sec)
```

Initial Car table

```
mysql> select * from owner;
```

Owner_Id
1
2
3
4
5
6
7
8
9
10

```
10 rows in set (0.01 sec)
```

Initial Owner Table

Initial Individual Customer table

```
mysql> select * from individual_cust;
+-----+-----+-----+
| Cust_ID | Initial_Name | LName |
+-----+-----+-----+
|      11 | Panini       | Pande |
|      22 | Animesh     | Gupta |
|      33 | Muskan      | Jain  |
|      44 | Karan       | Singh |
|      55 | Aditya      | Bhat  |
|      66 | Akash       | Biswas|
|      77 | Shreya      | Patel |
|      88 | Jatin       | S     |
|      99 | Harshit     | Sandu |
|     100 | Smit        | Joshi |
+-----+-----+-----+
10 rows in set (0.01 sec)
```

```
mysql> select * from rents;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Vehicle_Id | Cust_ID | Start_Date | Return_Date | No_of_Weeks | Daily_Rate | Weekly_Rate | No_of_days | Scheduled | Amount_Due |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|      1001 |      11 | 2023-04-01 | 2023-04-08 |           1 |        100 |        200 |           7 |           0 |         200 |
|      1002 |      22 | 2023-04-02 | 2023-04-09 |           1 |        110 |        220 |           7 |           0 |         220 |
|      1003 |      33 | 2023-04-03 | 2023-04-10 |           1 |        120 |        240 |           7 |           0 |         240 |
|      1004 |      44 | 2023-04-04 | 2023-04-11 |           1 |        130 |        260 |           7 |           0 |         260 |
|      1005 |      55 | 2023-04-05 | 2023-04-12 |           1 |        140 |        280 |           7 |           0 |         280 |
|      1006 |      66 | 2023-04-06 | 2023-04-13 |           1 |        150 |        300 |           7 |           0 |         300 |
|      1007 |      77 | 2023-04-07 | 2023-04-14 |           1 |        160 |        320 |           7 |           0 |         320 |
|      1008 |      88 | 2023-04-08 | 2023-04-15 |           1 |        170 |        340 |           7 |           0 |         340 |
|      1009 |      99 | 2023-04-09 | 2023-04-16 |           1 |        180 |        360 |           7 |           0 |         360 |
|      1010 |     100 | 2023-04-10 | 2023-04-17 |           1 |        190 |        380 |           7 |           0 |         380 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
10 rows in set (0.01 sec)
```

Initial Rents Table

```
mysql> select * from van;
+-----+
| Type_ID |
+-----+
|      1001 |
|      1002 |
|      1003 |
|      1004 |
|      1005 |
|      1006 |
|      1007 |
|      1008 |
|      1009 |
|      1010 |
+-----+
10 rows in set (0.01 sec)
```

Initial Van Table

**Task 3:**

SELECT

CASE

WHEN c.Owner\_Type = 'Individual' THEN CONCAT(c.Owner\_First\_Name, '',  
c.Owner\_Last\_Name)

WHEN c.Owner\_Type = 'Company' THEN c.Owner\_Company\_Name

END AS Owner\_Name,

CONCAT(cd.Car\_Classification, '', cd.Car\_Type) AS Car\_Type,

COUNT(\*) AS Car\_Count,

SUM(CASE WHEN r.Rental\_Type = 'Weekly' THEN r.Amount\_Due ELSE 0 END) AS  
Weekly\_Earnings,

SUM(CASE WHEN r.Rental\_Type = 'Weekly' THEN r.Amount\_Due ELSE 0 END) /  
COUNT(\*) AS Earnings\_Per\_Car

FROM

car c

JOIN car\_desc cd ON c.Type\_ID = cd.Type\_ID

JOIN rental r ON c.Vehicle\_ID = r.Vehicle\_ID

WHERE

r.Schedule = 1

AND r.Start\_Date <= NOW()

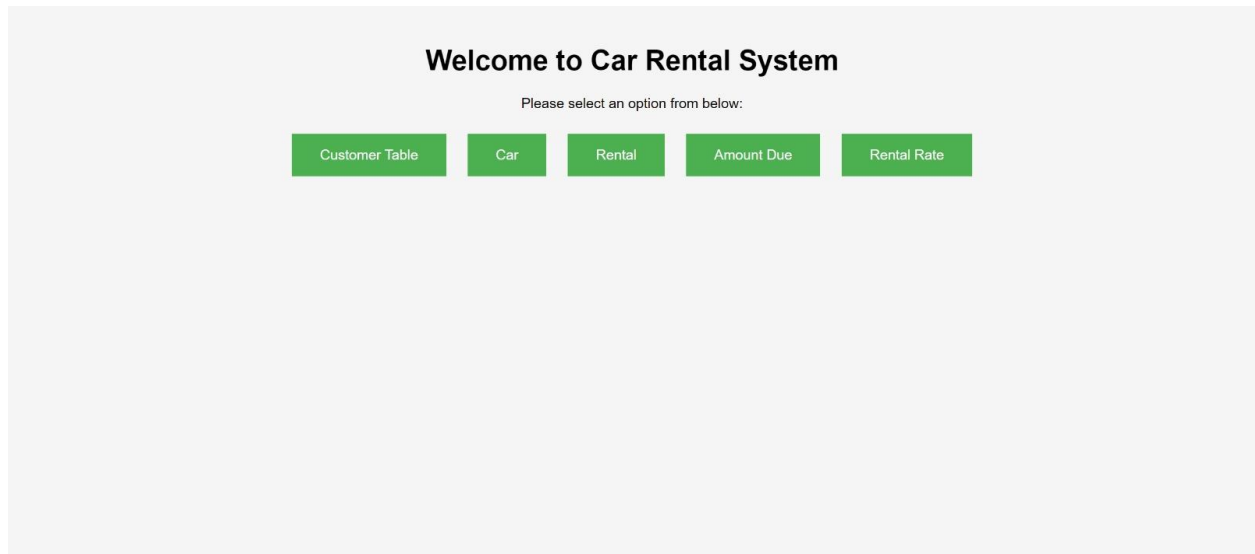
AND r.End\_Date >= NOW()

GROUP BY

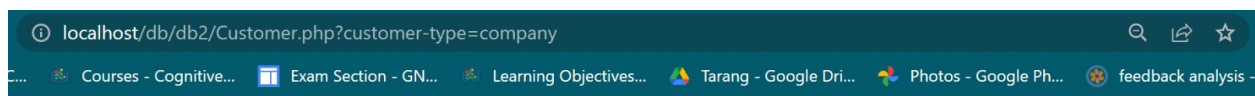
c.Owner\_ID,  
cd.Car\_Classification,  
cd.Car\_Type;

```
mysql> +-----+-----+-----+-----+-----+
-> | Owner_Id | Car_Type | Car_Count | Weekly_Earnings | Earnings_Per_Car |
-> +-----+-----+-----+-----+-----+
-> | 1        | Large   | 1         | 300              | 300              |
-> | 2        | Compact | 1         | 280              | 280              |
-> | 3        | Medium  | 1         | 260              | 260              |
-> +-----+-----+-----+-----+-----+
-> 3 rows in set (0.01 sec) _
```

**Task 4 and 5:** The following database update transactions were executed using a PHP based interface.



Initially, the user clicks on the "Customer Table" option from the homepage to proceed to the next form.

The image displays a web form for customer registration. The form is enclosed in a light gray border. It contains the following elements: a label "Customer ID" above a text input field containing "111"; a label "Phone Number" above a text input field containing "6693"; a row of two radio buttons, with "Company" selected (indicated by a blue dot) and "Individual" unselected; a label "Company Name" above a text input field containing "K"; and a blue "Submit" button at the bottom center of the form.

Then, the user will input the customer ID and phone number, and select the company by entering its name.

The changes are reflected on both the customer and company tables.

```
mysql> select * from customer;
```

+-----+-----+	
Cust_ID	Phone
+-----+-----+	
11	1234
22	2345
33	3456
44	4567
55	5678
66	6789
77	7890
88	8901
99	9012
100	1023
111	6693
+-----+-----+	

```
mysql> select * from company;
+-----+-----+
| Cust_ID | CName |
+-----+-----+
|      11 | A     |
|      22 | B     |
|      33 | C     |
|      44 | D     |
|      55 | E     |
|      66 | F     |
|      77 | G     |
|      88 | H     |
|      99 | I     |
|     100 | J     |
|     111 | K     |
+-----+-----+
11 rows in set (0.01 sec)
```

The user enters a new customer ID along with the individual's name, which includes the first and last name.



```
mysql> select * from customer;
```

Cust_ID	Phone
11	1234
22	2345
33	3456
44	4567
55	5678
66	6789
77	7890
88	8901
99	9012
100	1023
111	6693
112	6694

```
12 rows in set (0.01 sec)
```

```
mysql> select * from individual_cust;
```

Cust_ID	Initial_Name	LName
11	Panini	Pande
22	Animesh	Gupta
33	Muskan	Jain
44	Karan	Singh
55	Aditya	Bhat
66	Akash	Biswas
77	Shreya	Patel
88	Jatin	S
99	Harshit	Sandu
100	Smit	Joshi
112	Rudviq	Bhavsar

```
11 rows in set (0.01 sec)
```

The user will click on the "Car" option from the homepage to update the car details.

The screenshot shows a web browser window with the URL `localhost/db/db2/Car.php`. The browser's address bar and tabs are visible at the top. Below the browser window, a form is displayed with the following fields and values:

- Vehicle ID**: 1011
- Owner ID**: 11
- Start Date**: 11-04-2023
- End Date**: 12-04-2023
- Model Type**: Maruti
- Year**: 2023
- Car-Type**: Van
- Car-Category**: Regular

At the bottom of the form, there is a blue button labeled "Add Data".

Before insertion of new values

```
mysql> select * from car;
```

Vehicle_ID	Owner_ID	Avail_StartDate	Avail_EndDate	Model	Year
1001	1	2023-04-01	2023-04-02	Toyota	2014
1002	2	2023-04-02	2023-04-03	Nissan	2015
1003	3	2023-04-03	2023-04-04	Honda	2016
1004	4	2023-04-04	2023-04-05	Ford	2017
1005	5	2023-04-05	2023-04-06	Chevrolet	2018
1006	6	2023-04-06	2023-04-07	Volkswagen	2019
1007	7	2023-04-07	2023-04-08	Hyundai	2020
1008	8	2023-04-08	2023-04-09	Kia	2021
1009	9	2023-04-09	2023-04-10	MercedesBenz	2022
1010	10	2023-04-10	2023-04-11	BMW	2023

10 rows in set (0.02 sec)

```
mysql> select * from car_type  
-> ;
```

Type_ID	Daily_Rate	Weekly_Rate
1001	100	200
1002	110	220
1003	120	240
1004	130	260
1005	140	280
1006	150	300
1007	160	320
1008	170	340
1009	180	360
1010	190	380

10 rows in set (0.02 sec)

```
mysql> select * from van;
```

Type_ID
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010

```
10 rows in set (0.01 sec)
```

```
mysql> select * from company;
```

+-----+-----+	
Cust_ID   CName	
+-----+-----+	
11   A	
22   B	
33   C	
44   D	
55   E	
66   F	
77   G	
88   H	
99   I	
100   J	
+-----+-----+	

```
10 rows in set (0.01 sec)
```

```
mysql> select * from customer;
```

Cust_ID	Phone
11	1234
22	2345
33	3456
44	4567
55	5678
66	6789
77	7890
88	8901
99	9012
100	1023

```
10 rows in set (0.01 sec)
```

```
mysql> select * from rents;
```

Vehicle_Id	Cust_ID	Start_Date	Return_Date	No_of_Weeks	Daily_Rate	Weekly_Rate	No_of_days	Scheduled	Amount_Due
1001	11	2023-04-01	2023-04-08	1	100	200	7	0	200
1002	22	2023-04-02	2023-04-09	1	110	220	7	0	220
1003	33	2023-04-03	2023-04-10	1	120	240	7	0	240
1004	44	2023-04-04	2023-04-11	1	130	260	7	0	260
1005	55	2023-04-05	2023-04-12	1	140	280	7	0	280
1006	66	2023-04-06	2023-04-13	1	150	300	7	0	300
1007	77	2023-04-07	2023-04-14	1	160	320	7	0	320
1008	88	2023-04-08	2023-04-15	1	170	340	7	0	340
1009	99	2023-04-09	2023-04-16	1	180	360	7	0	360
1010	100	2023-04-10	2023-04-17	1	190	380	7	0	380

```
10 rows in set (0.01 sec)
```

After inserting new values

```
mysql> select * from regular;
```

```
+-----+
```

```
| Type_ID |
```

```
+-----+
```

```
|      1001 |
```

```
|      1002 |
```

```
|      1003 |
```

```
|      1004 |
```

```
|      1005 |
```

```
|      1006 |
```

```
|      1007 |
```

```
|      1008 |
```

```
|      1009 |
```

```
|      1010 |
```

```
|      1011 |
```

```
+-----+
```

```
11 rows in set (0.01 sec)
```

```
mysql>
```

```
mysql> select * from van;
```

Type_ID
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011

```
11 rows in set (0.02 sec)
```



```
mysql> select * from rents;
```

Vehicle_Id	Cust_ID	Start_Date	Return_Date	No_of_Weeks	Daily_Rate	Weekly_Rate	No_of_days	Scheduled	Amount_Due
1001	11	2023-04-01	2023-04-08	1	100	200	7	0	200
1002	22	2023-04-02	2023-04-09	1	110	220	7	0	220
1003	33	2023-04-03	2023-04-10	1	120	240	7	0	240
1004	44	2023-04-04	2023-04-11	1	130	260	7	0	260
1005	55	2023-04-05	2023-04-12	1	140	280	7	0	280
1006	66	2023-04-06	2023-04-13	1	150	300	7	0	300
1007	77	2023-04-07	2023-04-14	1	160	320	7	0	320
1008	88	2023-04-08	2023-04-15	1	170	340	7	0	340
1009	99	2023-04-09	2023-04-16	1	180	360	7	0	360
1010	100	2023-04-10	2023-04-17	1	190	380	7	0	380
1011	112	2023-04-11	2023-04-12	1	200	400	7	0	400

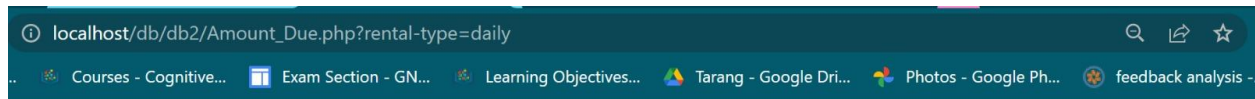
```
11 rows in set (0.01 sec)
```

```
mysql> select * from car;
```

Vehicle_ID	Owner_ID	Avail_StartDate	Avail_EndDate	Model	Year
1001	1	2023-04-01	2023-04-02	Toyota	2014
1002	2	2023-04-02	2023-04-03	Nissan	2015
1003	3	2023-04-03	2023-04-04	Honda	2016
1004	4	2023-04-04	2023-04-05	Ford	2017
1005	5	2023-04-05	2023-04-06	Chevrolet	2018
1006	6	2023-04-06	2023-04-07	Volkswagen	2019
1007	7	2023-04-07	2023-04-08	Hyundai	2020
1008	8	2023-04-08	2023-04-09	Kia	2021
1009	9	2023-04-09	2023-04-10	MercedesBenz	2022
1010	10	2023-04-10	2023-04-11	BMW	2023
1011	11	2023-04-11	2023-04-12	Maruti	2023

```
11 rows in set (0.01 sec)
```

The user click on amount due to find out the amount they have to pay for the rental car, which gets calculated in the backend and gets displayed in the frontend.



Vehicle ID  
1011

Customer ID  
112

Daily ☒

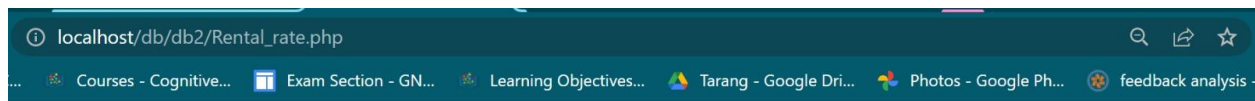
Number of Days  
1

Weekly ☐

Calculate

Total Amount Due  
200

After the user enters the number of days that the car has been rented for, the total amount due will be displayed as \$200 on the homepage.



Vehicle ID  
1011

Daily ☒

New Daily Rate  
300

Weekly ☐

UPDATE

Before inserting new value The daily rate value for vehicle ID 1011 will be updated from \$200 to \$300 :

```
mysql> select * from car_type  
-> ;
```

Type_ID	Daily_Rate	Weekly_Rate
1001	100	200
1002	110	220
1003	120	240
1004	130	260
1005	140	280
1006	150	300
1007	160	320
1008	170	340
1009	180	360
1010	190	380
1011	200	400

```
11 rows in set (0.01 sec)
```

After inserting new values

```
mysql> select * from rents;
```

Vehicle_Id	Cust_ID	Start_Date	Return_Date	No_of_Weeks	Daily_Rate	Weekly_Rate	No_of_days	Scheduled	Amount_Due
1001	11	2023-04-01	2023-04-08	1	100	200	7	0	200
1002	22	2023-04-02	2023-04-09	1	110	220	7	0	220
1003	33	2023-04-03	2023-04-10	1	120	240	7	0	240
1004	44	2023-04-04	2023-04-11	1	130	260	7	0	260
1005	55	2023-04-05	2023-04-12	1	140	280	7	0	280
1006	66	2023-04-06	2023-04-13	1	150	300	7	0	300
1007	77	2023-04-07	2023-04-14	1	160	320	7	0	320
1008	88	2023-04-08	2023-04-15	1	170	340	7	0	340
1009	99	2023-04-09	2023-04-16	1	180	360	7	0	360
1010	100	2023-04-10	2023-04-17	1	190	380	7	0	380
1011	112	2023-04-11	2023-04-12	1	300	400	7	0	400

11 rows in set (0.01 sec)

localhost/db/db2/Rental.php?rental-type=daily

Vehicle ID

1011

Customer ID

112

Car-Type

Van

Daily

Weekly

Start Date

11/04/2023

End Date

12/04/2023

Show Data

Available Cars

Maruti



