Car Rental Project: **Phase 2**

Task 1: Create the following 4 tables for the Car Rental Database

1. **CUSTOMER** table

```
CREATE TABLE CUSTOMER (
    CustID INTEGER NOT NULL,
    Name TEXT NOT NULL,
    Phone TEXT NOT NULL,
    PRIMARY KEY (CustID)
);
```

2. **RENTAL** table

```
CREATE TABLE RENTAL (
    CustID INTEGER NOT NULL,
    VehicleID TEXT NOT NULL,
    StartDate DATE NOT NULL,
    OrderDate DATE NOT NULL,

    RentalType INTEGER NOT NULL,
    Qty INTEGER NOT NULL,
    ReturnDate DATE NOT NULL,
    TotalAmount INTEGER NOT NULL,
    PaymentDate DATE,

    PRIMARY KEY (RentalID),
    FOREIGN KEY (CustID) REFERENCES CUSTOMER(CustID),
    FOREIGN KEY (VehicleID) REFERENCES Vehicle(VehicleID)
);
```

3. **VEHICLE** table

```
CREATE TABLE VEHICLE (
    VehicleID TEXT NOT NULL,
    Description TEXT NOT NULL,
    Year INTEGER NOT NULL,
    Type TEXT NOT NULL,
    Category INTEGER NOT NULL,

    PRIMARY KEY (VehicleID)
);
```

4. **RATE** table

```
CREATE TABLE RATE (
    Type INTEGER NOT NULL,
    Category INTEGER NOT NULL,
    Weekly INTEGER NOT NULL,
    Daily INTEGER NOT NULL,

    FOREIGN KEY (Type) REFERENCES VEHICLE(Type)
);
```

Task 2: Load the data from the text files into the corresponding tables

We ran the following commands to load the data from the text files into the corresponding tables.

```
.read taskone.sql ---this will create the tables from task 1
.mode csv
.import CUSTOMER.csv CUSTOMER
.import RENTAL.csv RENTAL
.import VEHICLE.csv VEHICLE
.import RATE.csv RATE
```

```
sqlite> SELECT COUNT(CustID) AS CUSTOMER_NUM FROM CUSTOMER;
CUSTOMER_NUM
------------
31
sqlite> SELECT COUNT(CustID) AS RENTAL_NUM FROM RENTAL;
RENTAL_NUM
----------
23
sqlite> SELECT COUNT(VehicleID) AS VEHICLE_NUM FROM VEHICLE;
VEHICLE_NUM
-----------
60
sqlite> SELECT COUNT(Type) AS RATE_NUM FROM RATE;
RATE_NUM
--------
10
```

Task 3: Then execute the following queries on the database tables:

Before:

```
.header on
.mode column
```

Question 1: Insert yourself as a New Customer. Do not provide the CustomerID in your query.

```
INSERT INTO CUSTOMER VALUES(NULL, 'Subash Bhusal', '(456) 7815-7884');
```

```
sqlite> INSERT INTO CUSTOMER VALUES(NULL, 'Subash Bhusal', '(456) 7815-7884');
sqlite> SELECT * FROM CUSTOMER;
CustID  Name              Phone
------  ----------------  ---------------
201     A. Parks          (214) 555-0127
202     S. Patel          (849) 811-6298
203     A. Hernandez      (355) 572-5385
204     G. Carver         (753) 763-8656
205     Sh. Byers         (912) 925-5332
206     L. Lutz           (931) 966-1775
207     L. Bernal         (884) 727-0591
208     I. Whyte          (811) 979-7345
209     L. Lott           (954) 706-2219
210     G. Clarkson       (309) 625-1838
211     Sh. Dunlap        (604) 581-6642
212     H. Gallegos       (961) 265-8638
213     L. Perkins        (317) 996-3104
214     M. Beach          (481) 422-0282
215     C. Pearce         (599) 881-5189
216     A. Hess           (516) 570-6411
217     M. Lee            (369) 898-6162
218     R. Booker         (730) 784-6303
219     A. Crowther       (325) 783-4081
220     H. Mahoney        (212) 262-8829
221     J. Brown          (644) 756-0110
222     H. Stokes         (931) 969-7317
223     J. Reeves         (940) 981-5113
224     A. Mcghee         (838) 610-5802
225     L. Mullen         (798) 331-7777
226     R. Armstrong      (325) 783-4081
227     J. Greenaway      (212) 262-8829
228     K. Kaiser Acosta  (228) 576-1557
229     D. Kirkpatrick    (773) 696-8009
230     A. Odonnell       (439) 536-8929
231     K. Kay            (368) 336-5403
232     Subash Bhusal     (456) 7815-7884
```

Question 2: Update your phone number to (837) 721-8965

```
UPDATE CUSTOMER SET Phone = '(837) 721-8965' WHERE Name = 'Subash Bhusal';
```

```
sqlite> UPDATE CUSTOMER SET Phone = '(837) 721-8965' WHERE Name = 'Subash Bhusal';
sqlite> SELECT * FROM CUSTOMER;
CustID  Name              Phone
------  ----------------  --------------
201     A. Parks          (214) 555-0127
202     S. Patel          (849) 811-6298
203     A. Hernandez      (355) 572-5385
204     G. Carver         (753) 763-8656
205     Sh. Byers         (912) 925-5332
206     L. Lutz           (931) 966-1775
207     L. Bernal         (884) 727-0591
208     I. Whyte          (811) 979-7345
209     L. Lott           (954) 706-2219
210     G. Clarkson       (309) 625-1838
211     Sh. Dunlap        (604) 581-6642
212     H. Gallegos       (961) 265-8638
213     L. Perkins        (317) 996-3104
214     M. Beach          (481) 422-0282
215     C. Pearce         (599) 881-5189
216     A. Hess           (516) 570-6411
217     M. Lee            (369) 898-6162
218     R. Booker         (730) 784-6303
219     A. Crowther       (325) 783-4081
220     H. Mahoney        (212) 262-8829
221     J. Brown          (644) 756-0110
222     H. Stokes         (931) 969-7317
223     J. Reeves         (940) 981-5113
224     A. Mcghee         (838) 610-5802
225     L. Mullen         (798) 331-7777
226     R. Armstrong      (325) 783-4081
227     J. Greenaway      (212) 262-8829
228     K. Kaiser Acosta  (228) 576-1557
229     D. Kirkpatrick    (773) 696-8009
230     A. Odonnell       (439) 536-8929
231     K. Kay            (368) 336-5403
232     Subash Bhusal     (837) 721-8965
```

Question 3: Increase only daily rates for luxury vehicles by 5%

```
UPDATE RATE SET Daily = Daily + (Daily * 0.05) WHERE Category = 1;
```

```
sqlite> UPDATE RATE SET Daily = Daily + (Daily * 0.05) WHERE Category = 1;
sqlite> SELECT * FROM RATE;
Type  Category  Weekly  Daily
----  --------  ------  ------
1     0         480     80
1     1         600     105
2     0         530     90
```

```
2     1         660     115.5
3     0         600     100
3     1         710     126
4     0         685     115
4     1         800     141.75
5     0         780     130
6     0         685     115
```

Question 4-a: Insert a new luxury van with the following info: Honda Odyssey 2019, vehicle id: 5FNRL6H58KB133711

```
INSERT INTO VEHICLE VALUES('5FNRL6H58KB133711', 'Honda Odyssey', 2019, 6, 1);
```

```
YV4940NB5F1191453  Volvo XC70              2015  4     1
5FNRL6H58KB133711  Honda Odyssey          2019  6     1
sqlite>
```

Question 4-b: You also need to insert the following rates:

5	1	900.00	150.00
6	1	800.00	135.00

```
INSERT INTO RATE VALUES(5, 1, 900.00, 150.00);
INSERT INTO RATE VALUES(6, 1, 800.00, 135.00);
```

```
# DONT FORGET TO CREATE AND ADD
```

Question 5: Return all Compact(1) & Luxury(1) vehicles that were available for rent from June 01, 2019 until June 20, 2019. List VechicleID as VIN, Description, year, and how many days have been rented so far. You need to change the weeks into days

```

```

Question 6: Return a list with the remaining balance for the customer with the id '221'. List customername, and the balance.

```

```

```
SELECT CUST.Name, SUM(R.TotalAmount)
FROM CUSTOMER AS CUST, RENTAL AS R
WHERE CUST.CustID = 221;
```

```
sqlite> SELECT CUST.Name, SUM(R.TotalAmount)
   ...> FROM CUSTOMER AS CUST, RENTAL AS R
   ...> WHERE CUST.CustID = R.CustID AND CUST.CustID = 221;
Name      SUM(R.TotalAmount)
--------  ------------------
J. Brown  16000
```

Question 7: Create a report that will return all vehicles. List the VehicleID as VIN, Description, Year, Type, Category, and Weekly and Daily rates. For the vehicle Type and Category, you need to use the SQL Case statement to substitute the numbers with text. Order your results based on Category (first Luxury and then Basic) and Type based on the Type number, not the text.

```
SELECT V.VehicleID, V.Description, V.Year, V.Type, V.Category,
    CASE R.Category
        WHEN 0 THEN 'Basic'
        WHEN 1 THEN 'Luxury'
    END AS Category, R.Weekly, R.Daily
FROM VEHICLE AS V, RATE AS R
WHERE V.Type = R.Type AND V.Category = R.Category
ORDER BY R.Category DESC;
```

```
--fill the commands
sqlite> SELECT V.VehicleID, V.Description, V.Year, V.Type, V.Category,
   ...>     CASE R.Category
   ...>         WHEN 0 THEN 'Basic'
   ...>         WHEN 1 THEN 'Luxury'
   ...>     END AS Category, R.Weekly, R.Daily
   ...> FROM VEHICLE AS V, RATE AS R
   ...> WHERE V.Type = R.Type AND V.Category = R.Category
   ...> ORDER BY R.Category DESC;
VehicleID          Description             Year  Type  Category  Category  Weekly
Daily
19VDE1F3XEE414842  Acura ILX               2014  1     1         Luxury    600
104.7375
1VWCH7A3XEC037969  Volkswagen Passat       2014  2     1         Luxury    660
115.21125
5N1AL0MM8EL549388  Infiniti JX35           2014  4     1         Luxury    800
141.395625
JH4KC1F50EC800004  Acura RLX               2014  3     1         Luxury    710
125.685
```

```
JH4KC1F56EC000095  Acura RLX               2014  3     1         Luxury    710
125.685
JTHBW1GG1F120DU53  Lexus ES 300h           2015  2     1         Luxury    660
115.21125
JTHCE1BL3F151DE04  Lexus GS 350            2015  2     1         Luxury    660
115.21125
JTHDL5EF9F5007221  Lexus LS 460            2015  3     1         Luxury    710
125.685
JTHFF2C26F135BX45  Lexus IS 250C           2015  1     1         Luxury    600
104.7375
JTJHY7AX2F120EA11  Lexus LX 570            2015  4     1         Luxury    800
141.395625
JTJJM7FX2E152CD75  Lexus GX460             2014  4     1         Luxury    800
141.395625
WA1LGAFE8ED001506  Audi Q7                 2014  4     1         Luxury    800
141.395625
WAU32AFD8FN005740  Audi A8                 2015  3     1         Luxury    710
125.685
WAUTFAFH0E0010613  Audi A5                 2014  1     1         Luxury    600
104.7375
WBA3A9G51ENN73366  BMW 3 Series            2014  1     1         Luxury    600
104.7375
WBA3B9C59EP458859  BMW 3 Series            2014  1     1         Luxury    600
104.7375
WBAVL1C57EVR93286  BMW X1                  2014  4     1         Luxury    800
141.395625
WDCGG0EB0EG188709  Mercedes_Benz GLK       2014  1     1         Luxury    600
104.7375
---some output ommited---
```

Question 8: What is the total of money that customers paid to us until today?

```
SELECT SUM(R.TotalAmount) AS TotalAmount_Paid
FROM RENTAL AS R
WHERE PaymentDate <= Date('now');
```

```
sqlite> SELECT SUM(R.TotalAmount) AS TotalAmount_Paid
   ...> FROM RENTAL AS R
   ...> WHERE PaymentDate <= Date('now');
TotalAmount_Paid
----------------
8230
```

Question 9-a: Create a report for the J. Brown customer with all vehicles he rented. List the description, year, type, and category. Also, calculate the unit price for every rental, the total duration mention if it is on weeks or days, the total amount, and if there is any payment.

Similarly, as in Question 7, you need to change the numeric values to the corresponding text. Order the results by the StartDate.

```
SELECT V.Description, V.Year,
CASE V.Type
    WHEN 1 THEN 'Compact'
    WHEN 2 THEN 'Midsize'
    WHEN 3 THEN 'Luxury'
    WHEN 4 THEN 'SUV'
    WHEN 5 THEN 'Truck'
    WHEN 6 THEN 'Van'
END AS Type,
CASE V.Category
    WHEN 0 THEN 'Basic'
    WHEN 1 THEN 'Luxury'
END AS Category,
R.TotalAmount/R.Qty AS 'Unit Price',
CASE R.RentalType
    WHEN 7 THEN ((JULIANDAY(R.ReturnDate)-JULIANDAY(R.StartDate))/7) || ' Weeks'
    WHEN 1 THEN (JULIANDAY(R.ReturnDate)-JULIANDAY(R.StartDate)) ||' Days'
END AS TimeRented,
R.TotalAmount,
CASE R.PaymentDate
    WHEN 'NULL' THEN 'Not Paid'
    ELSE 'Paid'
END AS Payment
FROM VEHICLE AS V, RENTAL AS R, CUSTOMER AS C
WHERE V.VehicleID = R.VehicleID AND R.CustID = 221 AND C.CustID = 221
ORDER BY R.StartDate;
```

```
Description         Year  Type     Category  Unit Price  TimeRented  TotalAmount
Payment
Acura ILX           2014  Compact  Luxury    600         1.0 Weeks   600
Paid
Audi A5             2014  Compact  Luxury    600         1.0 Weeks   600
Paid
Acura ILX           2014  Compact  Luxury    100         2.0 Days    200
Paid
Audi A5             2014  Compact  Luxury    100         2.0 Days    200
Paid
Acura ILX           2014  Compact  Luxury    600         4.0 Weeks   2400
Not Paid
Lexus IS 250C       2015  Compact  Luxury    600         4.0 Weeks   2400
Not Paid
Audi A5             2014  Compact  Luxury    600         4.0 Weeks   2400
Not Paid
BMW 3 Series        2014  Compact  Luxury    600         4.0 Weeks   2400
Not Paid
BMW 3 Series        2014  Compact  Luxury    600         4.0 Weeks   2400
Not Paid
```

```
Mercedes_Benz GLK  2014  Compact  Luxury    600         4.0 Weeks   2400
Not Paid
```