PROJECT 2 PART 2

CSE 3330
GROUP 8

HONOR CODE:
I pledge, on my honor, to uphold UT Arlington's tradition of academic integrity, a tradition that values hard work and honest effort in the pursuit of academic excellence. I promise that I will submit only work that I personally create or that I contribute to group collaborations, and I will appropriately reference any work from other sources. I will follow the highest standards of integrity and uphold the spirit of the Honor Code.

Shivam Patel

Gerald Kimeu

Carlos Uresti

Intro

In part 2 of the Rental Database, we will create tables corresponding to the relational schema, insert data into the table and run queries against that data. Task 1 will be creating the tables, task 2 will be loading the data and task 3 will be executing the queries specified by the requirements.

TASK 1

CUSTOMER

```
CREATE TABLE [CUSTOMER] (
  [CustID] INTEGER NOT NULL,
  [Name] varchar(45) NOT NULL,
  [Phone] varchar(50) NOT NULL,
  PRIMARY KEY ([CustID])
);
```

For the customer table we declared CustID as INTEGER instead of int so the CustID is automatically incremented by SQLITE3. Phone numbers are varchar instead of ints because they can have parentheses and dashes.

RATE

```
CREATE TABLE [RATE] (
  [Type] int NOT NULL,
  [Category] int NOT NULL,
  [Weekly] int NOT NULL,
  [Daily] int NOT NULL,
  PRIMARY KEY ([Type], [Category]),
  FOREIGN KEY ([Category]) REFERENCES [VEHICLE] ([Category]),
  FOREIGN KEY (Type) REFERENCES [VEHICLE] ([Type])
);
```

RENTAL

```
CREATE TABLE [RENTAL] (
    [CustID] int NOT NULL,
    [VehicleID] varchar(120) NOT NULL,
    [StartDate] date NOT NULL,
    [OrderDate] date NOT NULL,
    [RentalType] int NOT NULL,
    [Qty] int NOT NULL,
    [ReturnDate] date NOT NULL,
    [TotalAmount] int NOT NULL,
    [PaymentDate] date DEFAULT NULL,
    PRIMARY KEY ([CustID],[VehicleID],[StartDate]),
    FOREIGN KEY ([CustID]) REFERENCES [CUSTOMER] ([CustID]) ON DELETE
CASCADE ON UPDATE CASCADE,
    FOREIGN KEY ([VehicleID]) REFERENCES [VEHICLE] ([VehicleID]) ON DELETE
CASCADE ON UPDATE CASCADE
);
```

VEHICLE

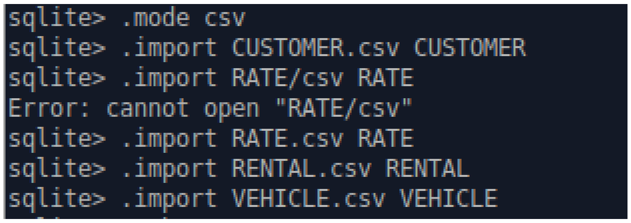
```
CREATE TABLE [VEHICLE] (
  [VehicleID] varchar(120) NOT NULL,
  [Description] varchar(50) NOT NULL,
  [Year] int NOT NULL,
  [Type] int NOT NULL,
  [Category] int NOT NULL,
  PRIMARY KEY ([VehicleID])
);

CREATE INDEX [IFK_RentalVehicleID] ON [RENTAL] ([VehicleID]);
CREATE INDEX [IFK_CustomerID] ON [RENTAL] ([CustID]);
CREATE INDEX [IFK_RateType] ON [RATE] ([Type]);
CREATE INDEX [IFK_RateCategory] ON [RATE] ([Category]);
```

TASK 2

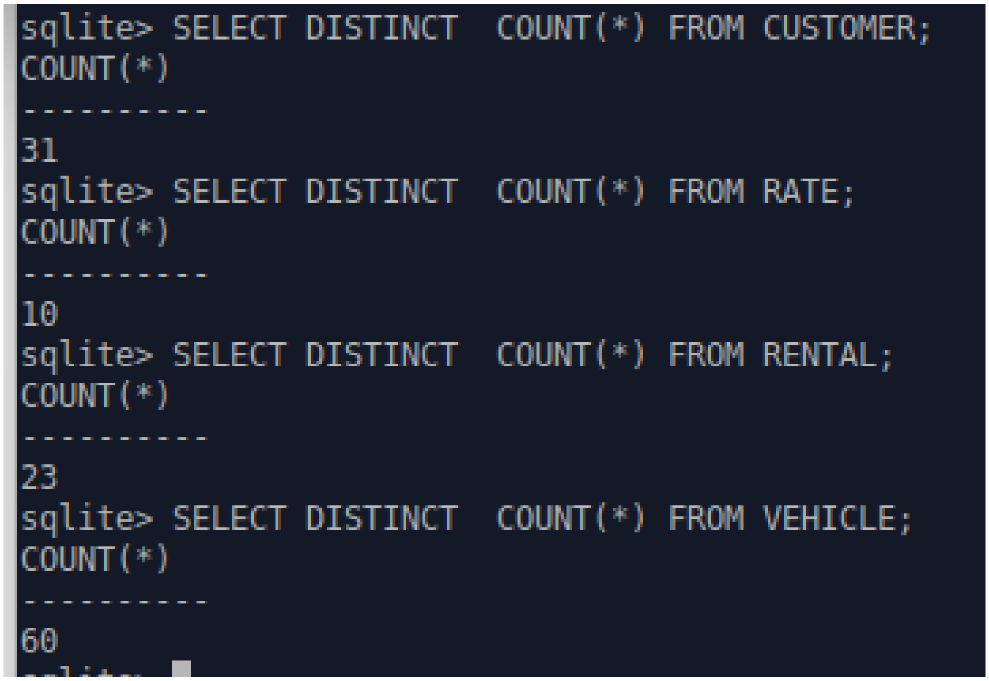
We populated the tables using the built-in .import function that sqlite3 includes which imports data from CSV files and populates them into the specified table and into the appropriate columns. The first command is “.mode CSV” which tells sqlite3 to delimit by commas.

Importing values into tables



```
sqlite> .mode csv
sqlite> .import CUSTOMER.csv CUSTOMER
sqlite> .import RATE/csv RATE
Error: cannot open "RATE/csv"
sqlite> .import RATE.csv RATE
sqlite> .import RENTAL.csv RENTAL
sqlite> .import VEHICLE.csv VEHICLE
```

Count for each table

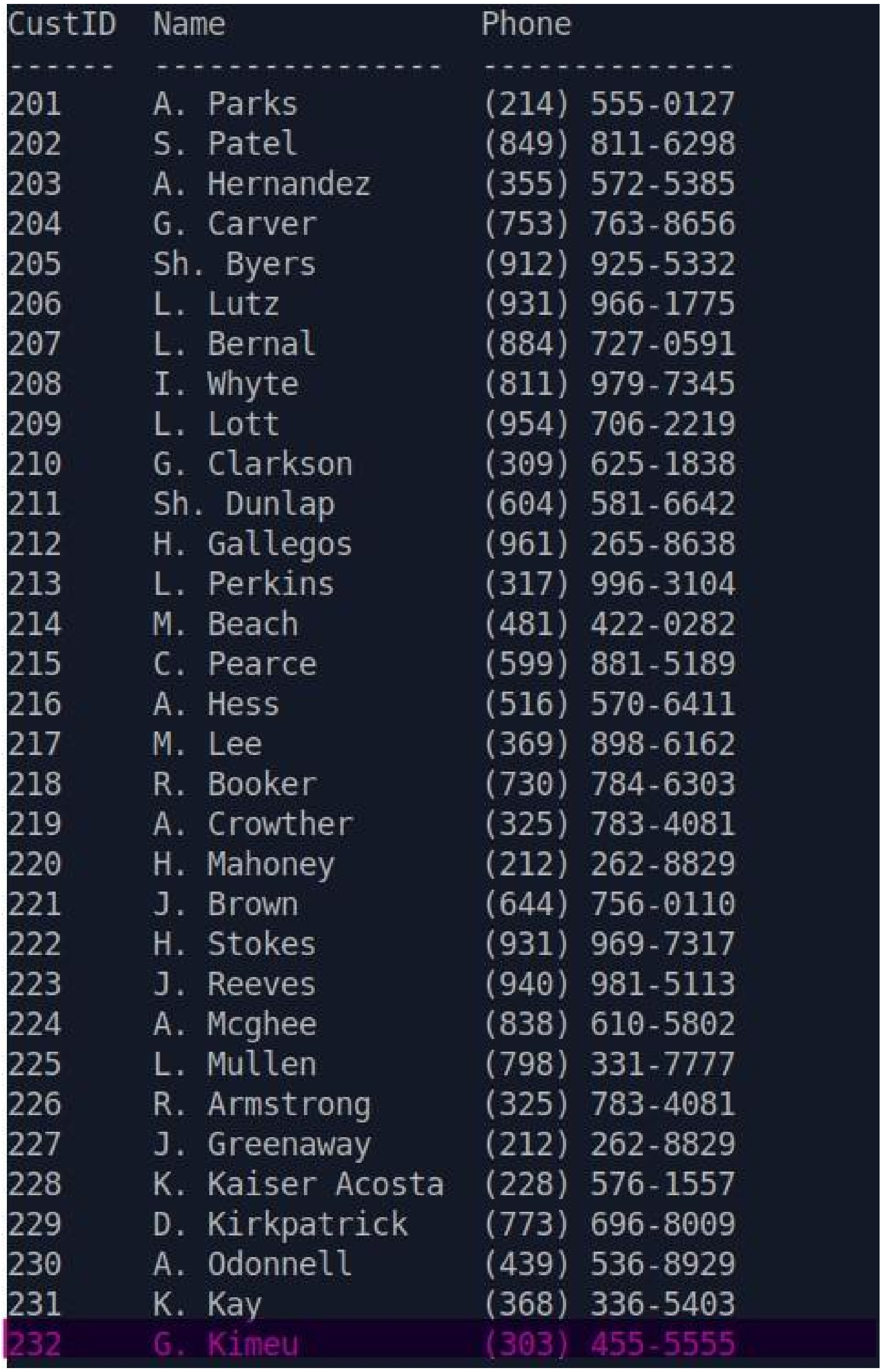


```
sqlite> SELECT DISTINCT  COUNT(*) FROM CUSTOMER;
COUNT(*)
----------
31
sqlite> SELECT DISTINCT  COUNT(*) FROM RATE;
COUNT(*)
----------
10
sqlite> SELECT DISTINCT  COUNT(*) FROM RENTAL;
COUNT(*)
----------
23
sqlite> SELECT DISTINCT  COUNT(*) FROM VEHICLE;
COUNT(*)
----------
60
```

TASK 3

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

INSERT into customer(Name,Phone) VALUES ('G. Kimeu', '(303) 455-5555');

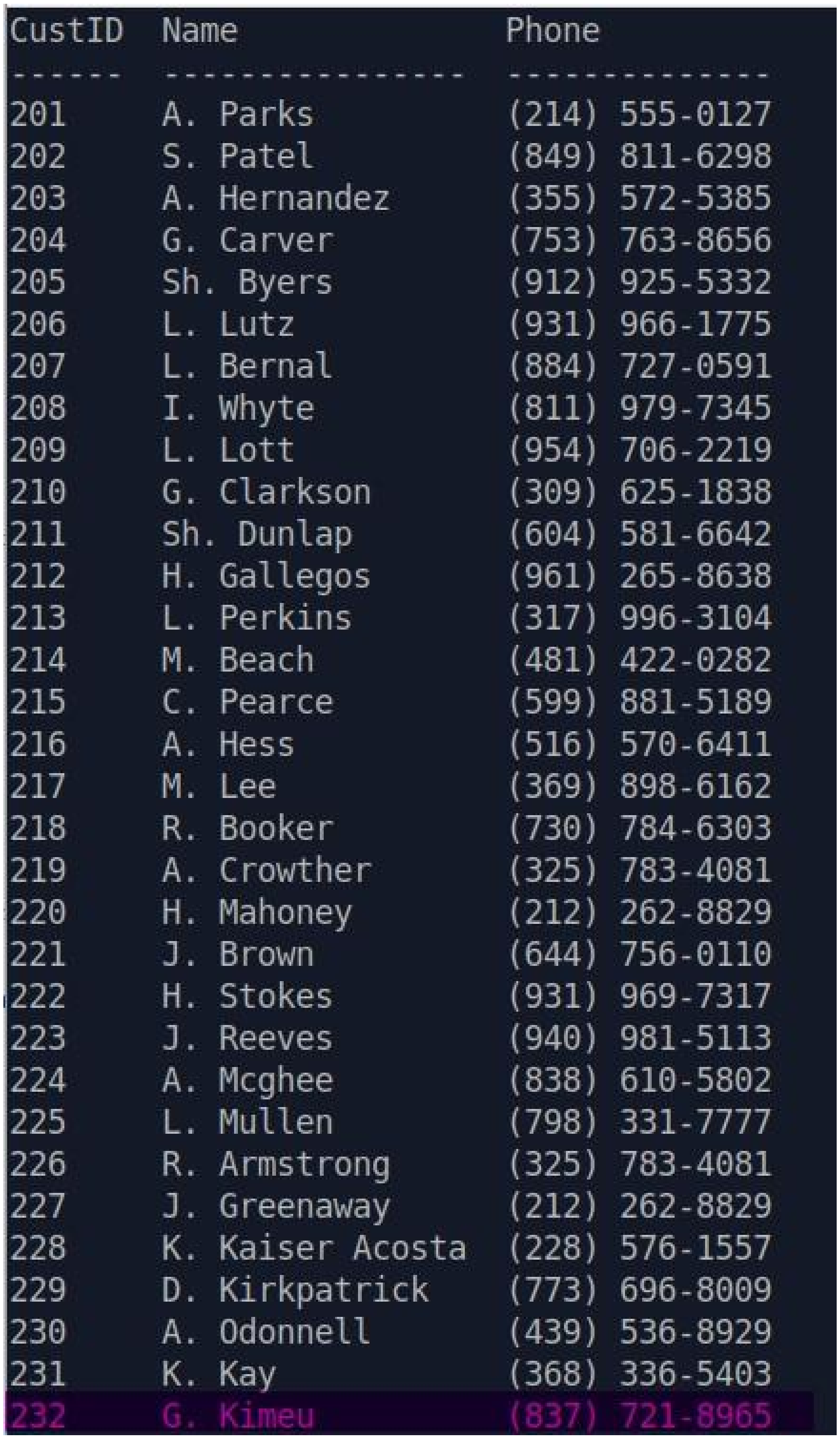


```
CustID  Name              Phone
------  ----------------  --------------
201     A. Parks          (214) 555-0127
202     S. Patel          (849) 811-6298
203     A. Hernandez      (355) 572-5385
204     G. Carver         (753) 763-8656
205     Sh. Byers         (912) 925-5332
206     L. Lutz           (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     G. Kimeu          (303) 455-5555
```

Result: 1 rows affected

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

UPDATE customer SET Phone = '(837) 721-8965' WHERE Name = 'G. Kimeu';

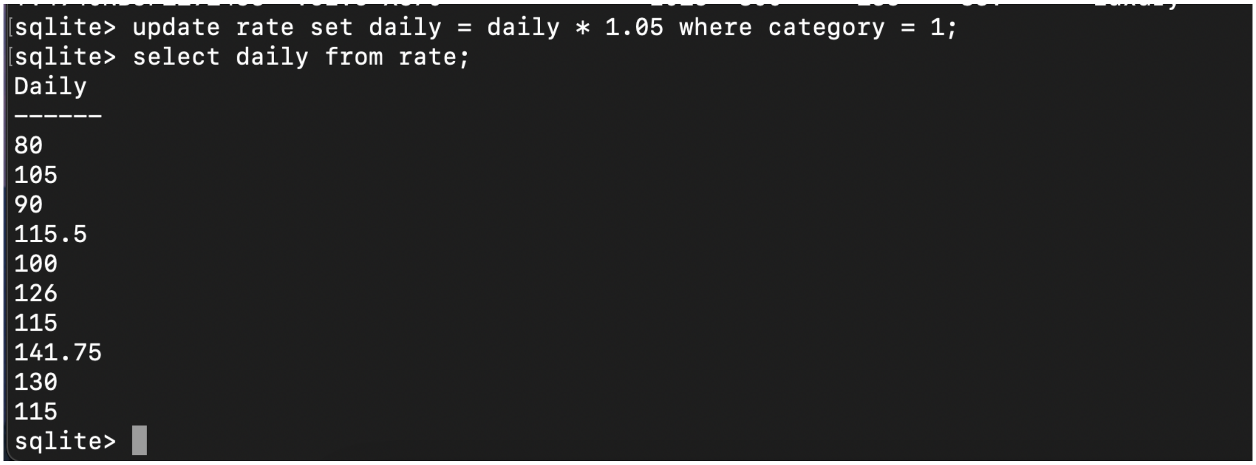


```
CustID  Name              Phone
------  ----------------  -------------
201     A. Parks          (214) 555-0127
202     S. Patel          (849) 811-6298
203     A. Hernandez      (355) 572-5385
204     G. Carver         (753) 763-8656
205     Sh. Byers         (912) 925-5332
206     L. Lutz           (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     G. Kimeu          (837) 721-8965
```

Result: 1 rows affected

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

update rate set daily = daily * 1.05 where category = 1;

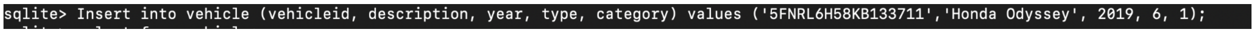


```
sqlite> update rate set daily = daily * 1.05 where category = 1;
sqlite> select daily from rate;
Daily
------
80
105
90
115.5
100
126
115
141.75
130
115
sqlite> 
```

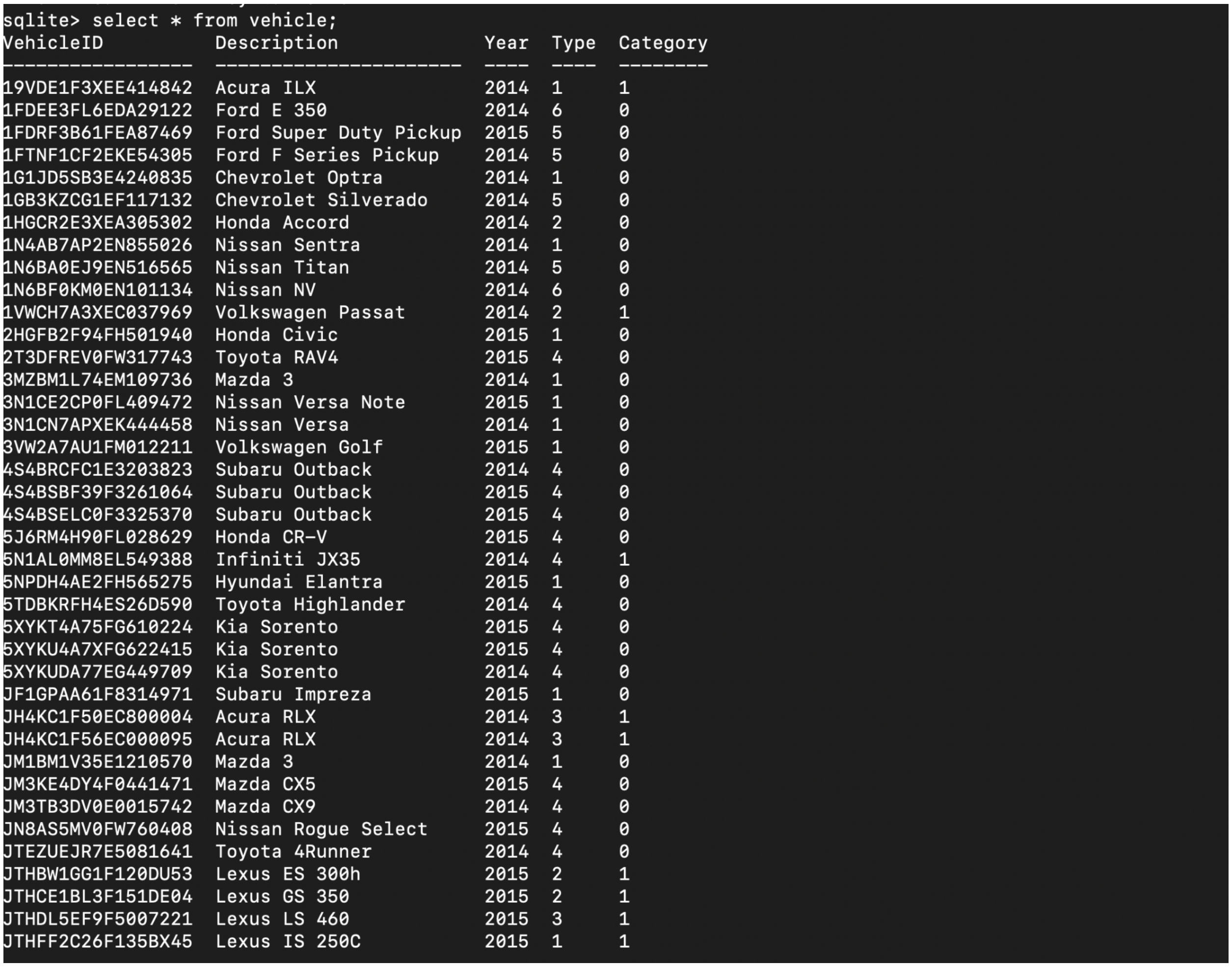
Result: 4 rows affected

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

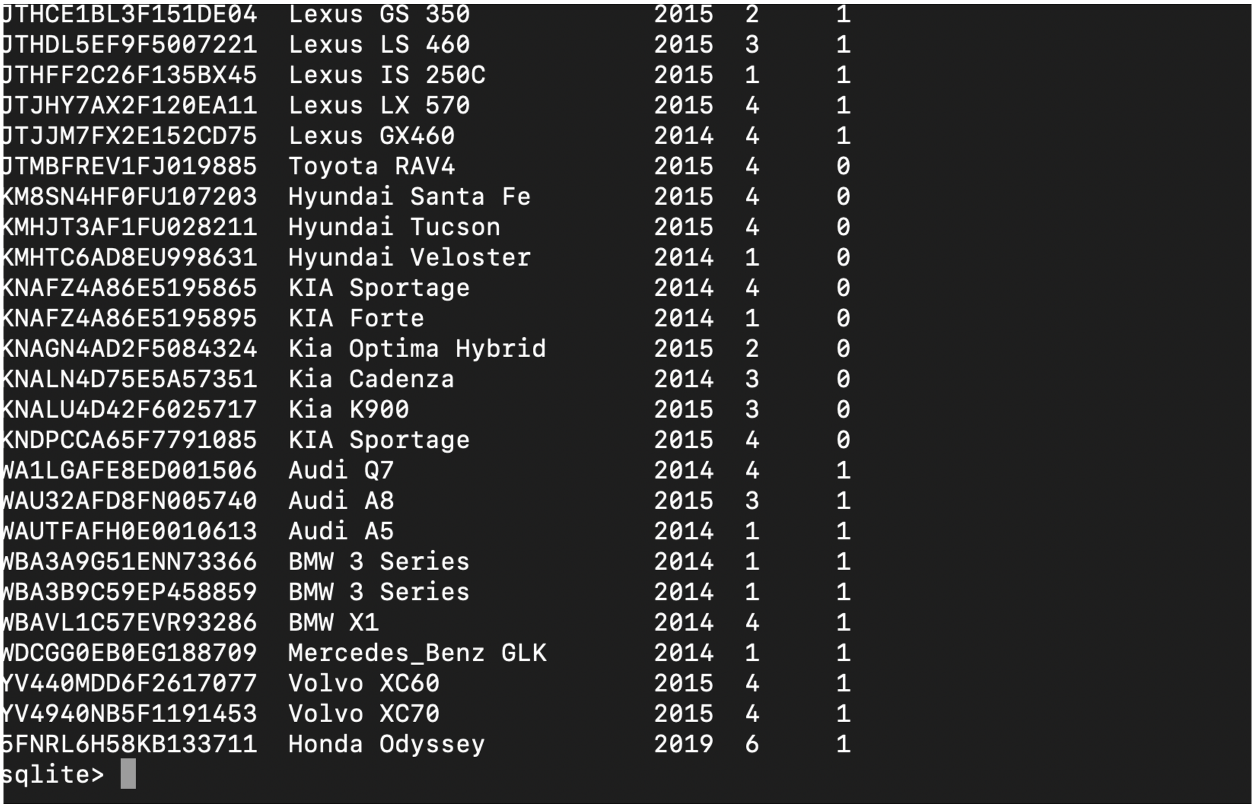
Insert into vehicle (vehicleid, description, year, type, category) values ('5FNRL6H58KB133711','Honda Odyssey', 2019, 6, 1);



```
sqlite> Insert into vehicle (vehicleid, description, year, type, category) values ('5FNRL6H58KB133711','Honda Odyssey', 2019, 6, 1);
```



```
sqlite> select * from vehicle;
VehicleID          Description             Year  Type  Category
-----------------  ----------------------  ----  ----  --------
19VDE1F3XEE414842  Acura ILX               2014  1     1
1FDEE3FL6EDA29122  Ford E 350              2014  6     0
1FDRF3B61FEA87469  Ford Super Duty Pickup  2015  5     0
1FTNF1CF2EKE54305  Ford F Series Pickup    2014  5     0
1G1JD5SB3E4240835  Chevrolet Optra         2014  1     0
1GB3KZCG1EF117132  Chevrolet Silverado     2014  5     0
1HGCR2E3XEA305302  Honda Accord            2014  2     0
1N4AB7AP2EN855026  Nissan Sentra           2014  1     0
1N6BA0EJ9EN516565  Nissan Titan            2014  5     0
1N6BF0KM0EN101134  Nissan NV               2014  6     0
1VWCH7A3XEC037969  Volkswagen Passat       2014  2     1
2HGFB2F94FH501940  Honda Civic             2015  1     0
2T3DFREV0FW317743  Toyota RAV4             2015  4     0
3MZBM1L74EM109736  Mazda 3                 2014  1     0
3N1CE2CP0FL409472  Nissan Versa Note       2015  1     0
3N1CN7APXEK444458  Nissan Versa            2014  1     0
3VW2A7AU1FM012211  Volkswagen Golf         2015  1     0
4S4BRCFC1E3203823  Subaru Outback          2014  4     0
4S4BSBF39F3261064  Subaru Outback          2015  4     0
4S4BSELC0F3325370  Subaru Outback          2015  4     0
5J6RM4H90FL028629  Honda CR-V              2015  4     0
5N1AL0MM8EL549388  Infiniti JX35           2014  4     1
5NPDH4AE2FH565275  Hyundai Elantra         2015  1     0
5TDBKRFH4ES26D590  Toyota Highlander       2014  4     0
5XYKT4A75FG610224  Kia Sorento             2015  4     0
5XYKU4A7XFG622415  Kia Sorento             2015  4     0
5XYKUDA77EG449709  Kia Sorento             2014  4     0
JF1GPAA61F8314971  Subaru Impreza          2015  1     0
JH4KC1F50EC800004  Acura RLX               2014  3     1
JH4KC1F56EC000095  Acura RLX               2014  3     1
JM1BM1V35E1210570  Mazda 3                 2014  1     0
JM3KE4DY4F0441471  Mazda CX5               2015  4     0
JM3TB3DV0E0015742  Mazda CX9               2014  4     0
JN8AS5MV0FW760408  Nissan Rogue Select     2015  4     0
JTEZUEJR7E5081641  Toyota 4Runner          2014  4     0
JTHBW1GG1F120DU53  Lexus ES 300h           2015  2     1
JTHCE1BL3F151DE04  Lexus GS 350            2015  2     1
JTHDL5EF9F5007221  Lexus LS 460            2015  3     1
JTHFF2C26F135BX45  Lexus IS 250C           2015  1     1
```

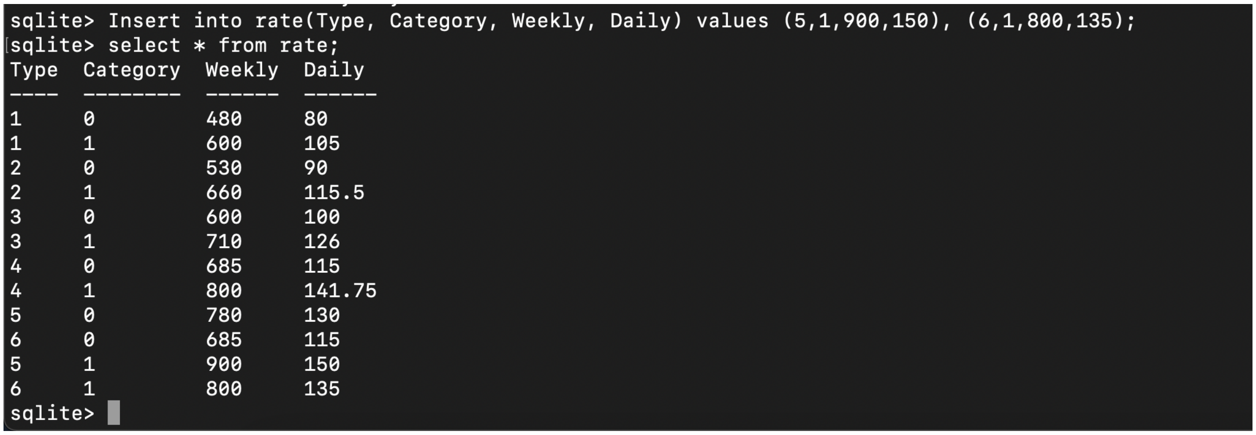


```
JTHCE1BL3F151DE04  Lexus GS 350           2015  2     1
JTHDL5EF9F5007221  Lexus LS 460           2015  3     1
JTHFF2C26F135BX45  Lexus IS 250C          2015  1     1
JTJHY7AX2F120EA11  Lexus LX 570           2015  4     1
JTJJM7FX2E152CD75  Lexus GX460            2014  4     1
JTMBFREV1FJ019885  Toyota RAV4            2015  4     0
KM8SN4HF0FU107203  Hyundai Santa Fe       2015  4     0
KMHJT3AF1FU028211  Hyundai Tucson         2015  4     0
KMHTC6AD8EU998631  Hyundai Veloster       2014  1     0
KNAFZ4A86E5195865  KIA Sportage           2014  4     0
KNAFZ4A86E5195895  KIA Forte              2014  1     0
KNAGN4AD2F5084324  Kia Optima Hybrid      2015  2     0
KNALN4D75E5A57351  Kia Cadenza            2014  3     0
KNALU4D42F6025717  Kia K900               2015  3     0
KNDPCCA65F7791085  KIA Sportage           2015  4     0
WA1LGAFE8ED001506  Audi Q7                2014  4     1
WAU32AFD8FN005740  Audi A8                2015  3     1
WAUTFAFH0E0010613  Audi A5                2014  1     1
WBA3A9G51ENN73366  BMW 3 Series           2014  1     1
WBA3B9C59EP458859  BMW 3 Series           2014  1     1
WBAVL1C57EVR93286  BMW X1                 2014  4     1
WDCGG0EB0EG188709  Mercedes_Benz GLK      2014  1     1
YV440MDD6F2617077  Volvo XC60             2015  4     1
YV4940NB5F1191453  Volvo XC70             2015  4     1
5FNRL6H58KB133711  Honda Odyssey          2019  6     1
sqlite>
```

Result: 1 rows affected

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

Insert into rate(Type, Category, Weekly, Daily) values (5,1,900,150), (6,1,800,135);

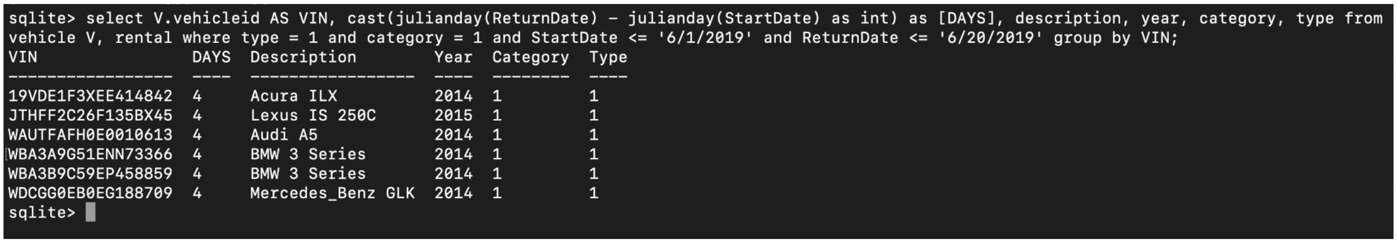


```
sqlite> Insert into rate(Type, Category, Weekly, Daily) values (5,1,900,150), (6,1,800,135);
[sqlite> select * from rate;
Type  Category  Weekly  Daily
----  --------  ------  ------
1     0         480     80
1     1         600     105
2     0         530     90
2     1         660     115.5
3     0         600     100
3     1         710     126
4     0         685     115
4     1         800     141.75
5     0         780     130
6     0         685     115
5     1         900     150
6     1         800     135
sqlite>
```

Result: 2 rows affected

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.

select V.vehicleid AS VIN, cast(julianday(ReturnDate) - julianday(StartDate) as int) as [DAYS], description, year, category, type from vehicle V, rental where type = 1 and category = 1 and StartDate <= '6/1/2019' and ReturnDate <= '6/20/2019' group by VIN;

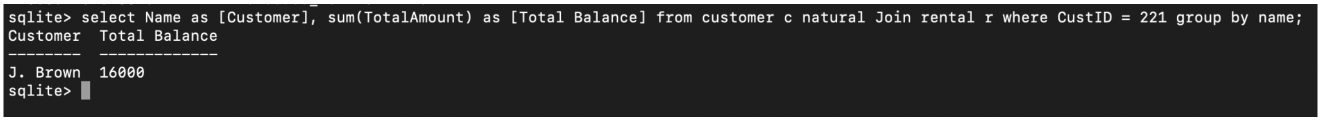


```
sqlite> select V.vehicleid AS VIN, cast(julianday(ReturnDate) - julianday(StartDate) as int) as [DAYS], description, year, category, type from
vehicle V, rental where type = 1 and category = 1 and StartDate <= '6/1/2019' and ReturnDate <= '6/20/2019' group by VIN;
VIN                DAYS  Description        Year  Category  Type
-----------------  ----  -----------------  ----  --------  ----
19VDE1F3XEE414842  4     Acura ILX          2014  1         1
JTHFF2C26F135BX45  4     Lexus IS 250C      2015  1         1
WAUTFAFH0E0010613  4     Audi A5            2014  1         1
WBA3A9G51ENN73366  4     BMW 3 Series       2014  1         1
WBA3B9C59EP458859  4     BMW 3 Series       2014  1         1
WDCGG0EB0EG188709  4     Mercedes_Benz GLK  2014  1         1
sqlite>
```

Result: 6 rows returned

Question 6: Return a list with the remaining balance for the customer with the id ‘221’. List customer name, and the balance.

select Name as [Customer], sum(TotalAmount) as [Total Balance] from customer c natural Join rental r where CustID = 221 group by name;



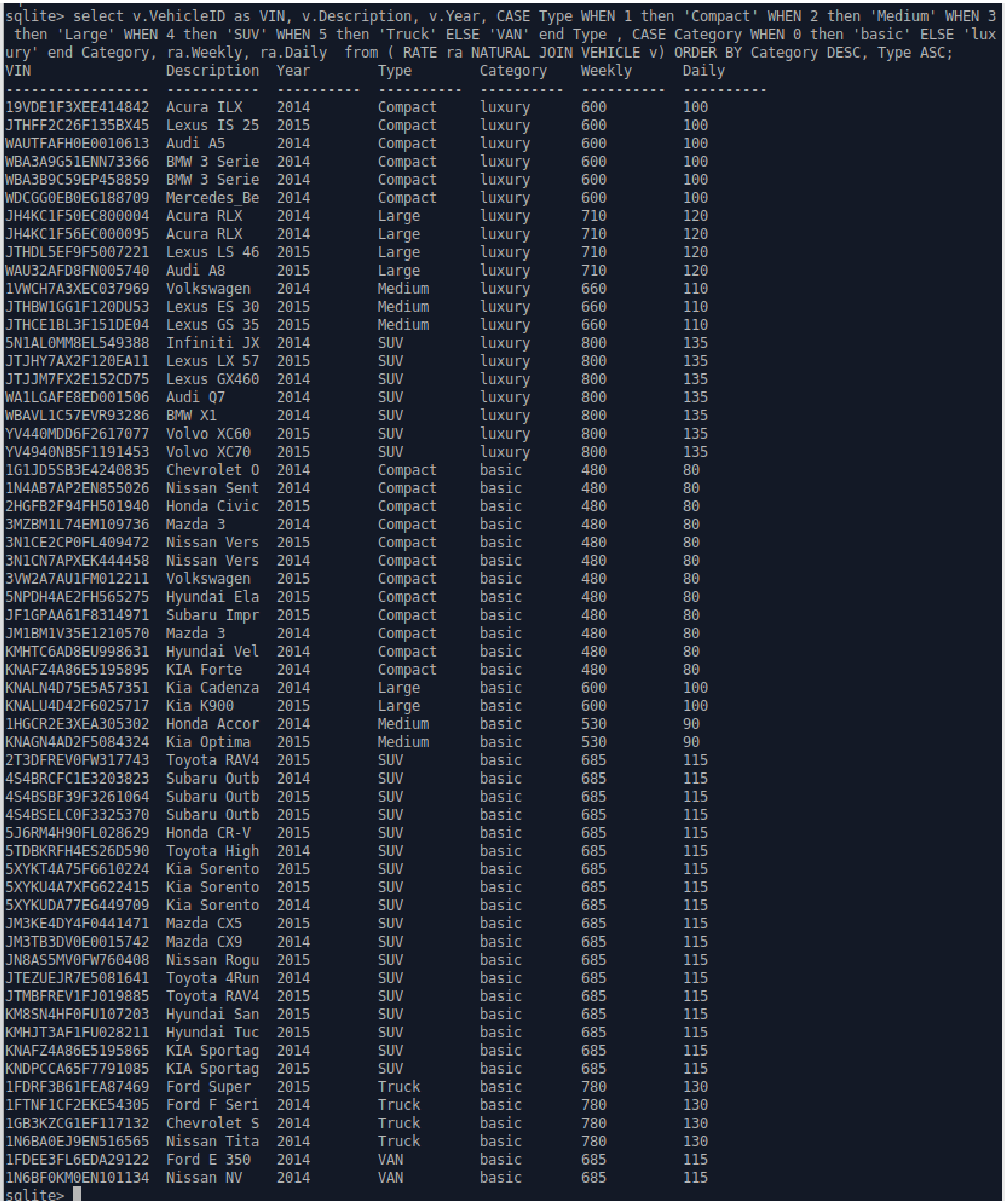
```
sqlite> select Name as [Customer], sum(TotalAmount) as [Total Balance] from customer c natural Join rental r where CustID = 221 group by name;
Customer  Total Balance
--------  -------------
J. Brown  16000
sqlite>
```

Result: 1 row returned

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 as VIN, v.Description, v.Year, CASE Type WHEN 1 then 'Compact' WHEN 2 then 'Medium' WHEN 3 then 'Large' WHEN 4 then 'SUV' WHEN 5 then 'Truck' ELSE 'VAN' end Type , CASE Category WHEN 0 then 'basic' ELSE 'luxury' end Category, ra.Weekly, ra.Daily from (RATE ra NATURAL JOIN VEHICLE v) ORDER BY Category DESC, Type ASC;

Queri

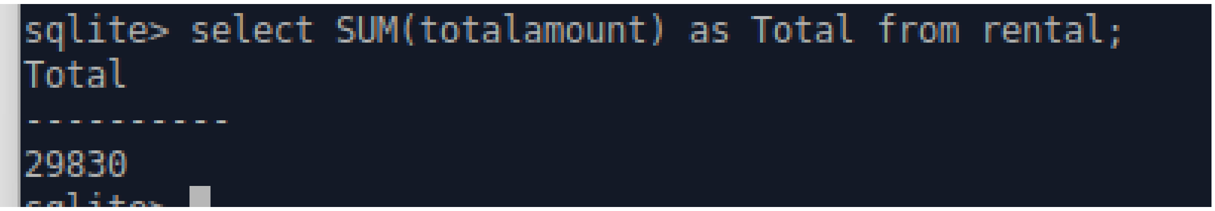


```
sqlite> select v.VehicleID as VIN, v.Description, v.Year, CASE Type WHEN 1 then 'Compact' WHEN 2 then 'Medium' WHEN 3
 then 'Large' WHEN 4 then 'SUV' WHEN 5 then 'Truck' ELSE 'VAN' end Type , CASE Category WHEN 0 then 'basic' ELSE 'lux
ury' end Category, ra.Weekly, ra.Daily  from ( RATE ra NATURAL JOIN VEHICLE v) ORDER BY Category DESC, Type ASC;
VIN                Description  Year        Type        Category    Weekly      Daily
-----------------  -----------  ----------  ----------  ----------  ----------  ----------
19VDE1F3XEE414842  Acura ILX    2014        Compact     luxury      600         100
JTHFF2C26F135BX45  Lexus IS 25  2015        Compact     luxury      600         100
WAUTFAFH0E0010613  Audi A5      2014        Compact     luxury      600         100
WBA3A9G51ENN73366  BMW 3 Serie  2014        Compact     luxury      600         100
WBA3B9C59EP458859  BMW 3 Serie  2014        Compact     luxury      600         100
WDCGG0EB0EG188709  Mercedes_Be  2014        Compact     luxury      600         100
JH4KC1F50EC800004  Acura RLX    2014        Large       luxury      710         120
JH4KC1F56EC000095  Acura RLX    2014        Large       luxury      710         120
JTHDL5EF9F5007221  Lexus LS 46  2015        Large       luxury      710         120
WAU32AFD8FN005740  Audi A8      2015        Large       luxury      710         120
1VWCH7A3XEC037969  Volkswagen   2014        Medium      luxury      660         110
JTHBW1GG1F120DU53  Lexus ES 30  2015        Medium      luxury      660         110
JTHCE1BL3F151DE04  Lexus GS 35  2015        Medium      luxury      660         110
5N1AL0MM8EL549388  Infiniti JX  2014        SUV         luxury      800         135
JTJHY7AX2F120EA11  Lexus LX 57  2015        SUV         luxury      800         135
JTJJM7FX2E152CD75  Lexus GX460  2014        SUV         luxury      800         135
WA1LGAFE8ED001506  Audi Q7      2014        SUV         luxury      800         135
WBAVL1C57EVR93286  BMW X1       2014        SUV         luxury      800         135
YV440MDD6F2617077  Volvo XC60   2015        SUV         luxury      800         135
YV4940NB5F1191453  Volvo XC70   2015        SUV         luxury      800         135
1G1JD5SB3E4240835  Chevrolet O  2014        Compact     basic       480         80
1N4AB7AP2EN855026  Nissan Sent  2014        Compact     basic       480         80
2HGFB2F94FH501940  Honda Civic  2015        Compact     basic       480         80
3MZBM1L74EM109736  Mazda 3      2014        Compact     basic       480         80
3N1CE2CP0FL409472  Nissan Vers  2015        Compact     basic       480         80
3N1CN7APXEK444458  Nissan Vers  2014        Compact     basic       480         80
3VW2A7AU1FM012211  Volkswagen   2015        Compact     basic       480         80
5NPDH4AE2FH565275  Hyundai Ela  2015        Compact     basic       480         80
JF1GPAA61F8314971  Subaru Impr  2015        Compact     basic       480         80
JM1BM1V35E1210570  Mazda 3      2014        Compact     basic       480         80
KMHTC6AD8EU998631  Hyundai Vel  2014        Compact     basic       480         80
KNAFZ4A86E5195895  KIA Forte    2014        Compact     basic       480         80
KNALN4D75E5A57351  Kia Cadenza  2014        Large       basic       600         100
KNALU4D42F6025717  Kia K900     2015        Large       basic       600         100
1HGCR2E3XEA305302  Honda Accor  2014        Medium      basic       530         90
KNAGN4AD2F5084324  Kia Optima   2015        Medium      basic       530         90
2T3DFREV0FW317743  Toyota RAV4  2015        SUV         basic       685         115
4S4BRCFC1E3203823  Subaru Outb  2014        SUV         basic       685         115
4S4BSBF39F3261064  Subaru Outb  2015        SUV         basic       685         115
4S4BSELC0F3325370  Subaru Outb  2015        SUV         basic       685         115
5J6RM4H90FL028629  Honda CR-V   2015        SUV         basic       685         115
5TDBKRFH4ES26D590  Toyota High  2014        SUV         basic       685         115
5XYKT4A75FG610224  Kia Sorento  2015        SUV         basic       685         115
5XYKU4A7XFG622415  Kia Sorento  2015        SUV         basic       685         115
5XYKUDA77EG449709  Kia Sorento  2014        SUV         basic       685         115
JM3KE4DY4F0441471  Mazda CX5    2015        SUV         basic       685         115
JM3TB3DV0E0015742  Mazda CX9    2014        SUV         basic       685         115
JN8AS5MV0FW760408  Nissan Rogu  2015        SUV         basic       685         115
JTEZUEJR7E5081641  Toyota 4Run  2014        SUV         basic       685         115
JTMBFREV1FJ019885  Toyota RAV4  2015        SUV         basic       685         115
KM8SN4HF0FU107203  Hyundai San  2015        SUV         basic       685         115
KMHJT3AF1FU028211  Hyundai Tuc  2015        SUV         basic       685         115
KNAFZ4A86E5195865  KIA Sportag  2014        SUV         basic       685         115
KNDPCCA65F7791085  KIA Sportag  2015        SUV         basic       685         115
1FDRF3B61FEA87469  Ford Super   2015        Truck       basic       780         130
1FTNF1CF2EKE54305  Ford F Seri  2014        Truck       basic       780         130
1GB3KZCG1EF117132  Chevrolet S  2014        Truck       basic       780         130
1N6BA0EJ9EN516565  Nissan Tita  2014        Truck       basic       780         130
1FDEE3FL6EDA29122  Ford E 350   2014        VAN         basic       685         115
1N6BF0KM0EN101134  Nissan NV    2014        VAN         basic       685         115
sqlite> 
```

Result: 60 rows returned

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

select SUM(totalamount) as Total from rental;

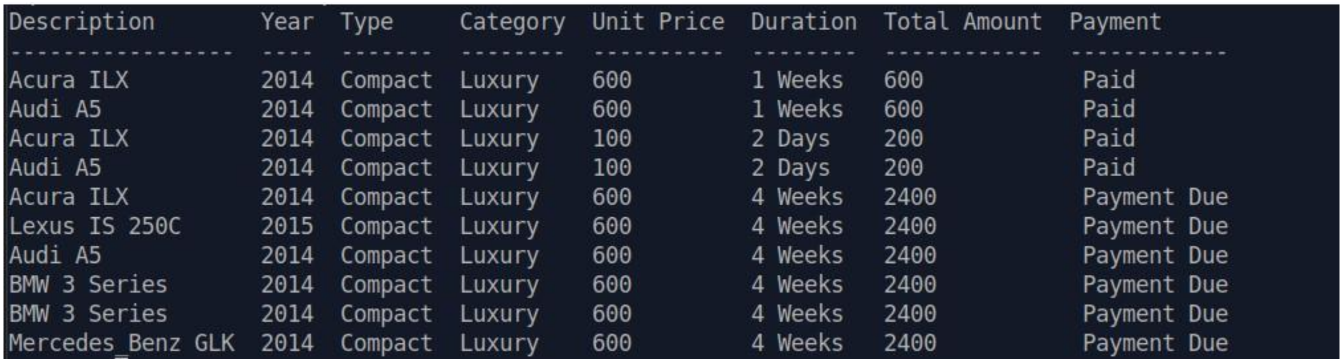


```
sqlite> select SUM(totalamount) as Total from rental;
Total
----------
29830
sqlite>
```

Result: 1 rows returned

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 Description, Year, CASE Type WHEN 1 then 'Compact'
WHEN 2 then 'Medium' WHEN 3 then 'Large'
WHEN 4 then 'SUV' WHEN 5 then 'Truck' ELSE 'VAN' end Type,
CASE Category WHEN 0 then 'Basic' ELSE 'Luxury' end Category,
cast(TotalAmount / Qty as int) as [Unit Price],
CASE RentalType WHEN 1 then cast(julianday(ReturnDate) - julianday(StartDate) as int) || ' Days' ELSE cast((julianday(ReturnDate) - julianday(StartDate)) / 7 as int) || ' Weeks' End Duration,TotalAmount as [Total Amount],
CASE PaymentDate WHEN 'NULL' then ' Payment Due' ELSE ' Paid' End Payment
from (customer c NATURAL join rental r) NATURAL join vehicle v where name = 'J. Brown' order by StartDate;

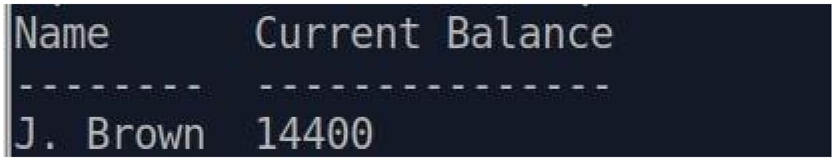


```
Description        Year  Type     Category  Unit Price  Duration  Total Amount  Payment
-----------------  ----  -------  --------  ----------  --------  ------------  ------------
Acura ILX          2014  Compact  Luxury    600         1 Weeks   600            Paid
Audi A5            2014  Compact  Luxury    600         1 Weeks   600            Paid
Acura ILX          2014  Compact  Luxury    100         2 Days    200            Paid
Audi A5            2014  Compact  Luxury    100         2 Days    200            Paid
Acura ILX          2014  Compact  Luxury    600         4 Weeks   2400           Payment Due
Lexus IS 250C      2015  Compact  Luxury    600         4 Weeks   2400           Payment Due
Audi A5            2014  Compact  Luxury    600         4 Weeks   2400           Payment Due
BMW 3 Series       2014  Compact  Luxury    600         4 Weeks   2400           Payment Due
BMW 3 Series       2014  Compact  Luxury    600         4 Weeks   2400           Payment Due
Mercedes_Benz GLK  2014  Compact  Luxury    600         4 Weeks   2400           Payment Due
```

Result: 10 rows returned

This query was challenging. It took us a while to get the unit price for the particular customer.

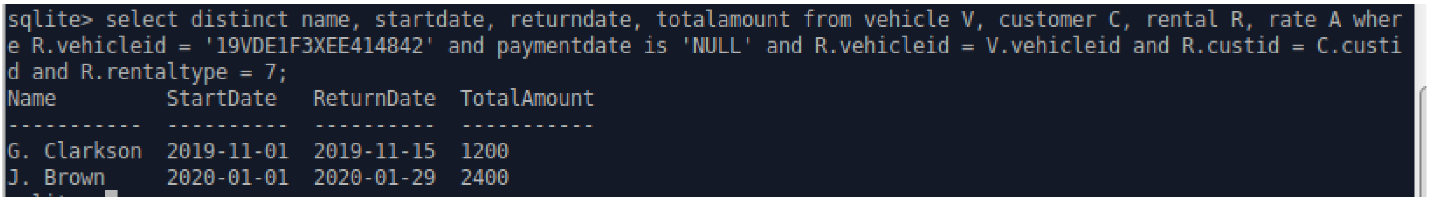
Question 9-b: For the same customer return the current balance.
select sum(TotalAmount) as [Current Balance]
from customer c NATURAL join rental where name = 'J. Brown' and PaymentDate = 'NULL';



```
Name      Current Balance
--------  ---------------
J. Brown  14400
```

Result: 1 rows returned

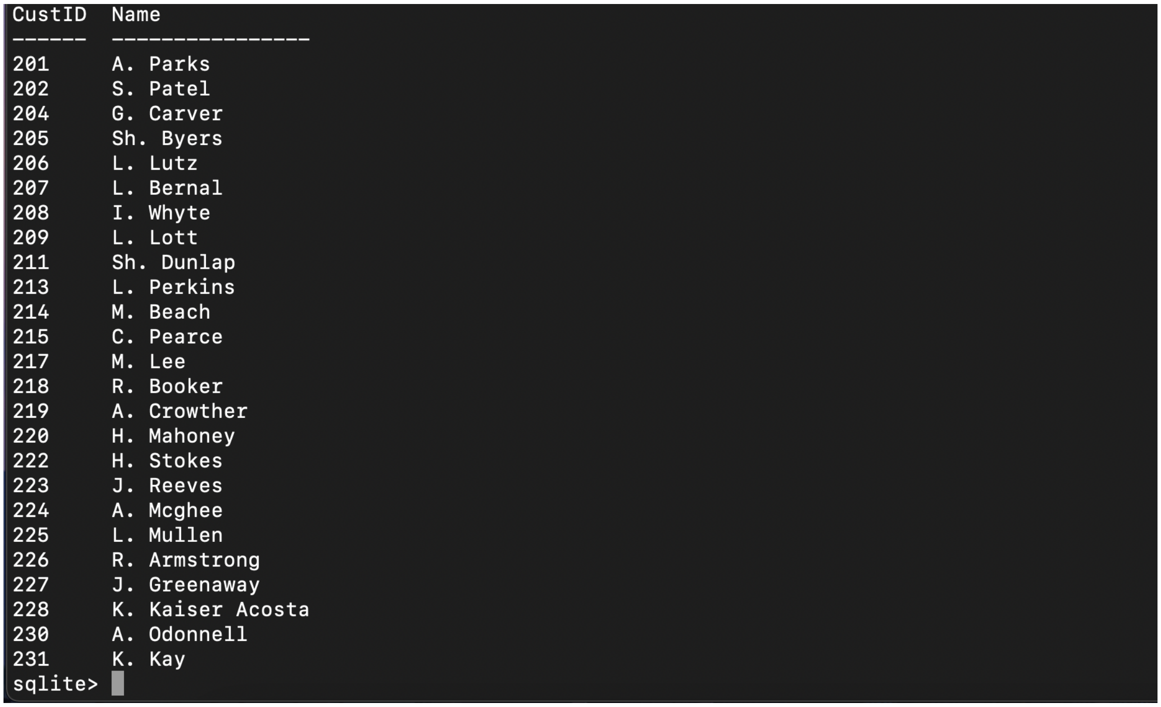
Question 10: Retrieve all weekly rentals for the vechicleID ‘19VDE1F3XEE414842’ that are not paid yet. List the Customer Name, the start and return date, and the amount.
select distinct name, startdate, returndate, totalamount from vehicle V, customer C, rental R, rate A where R.vehicleid = '19VDE1F3XEE414842' and paymentdate is 'NULL' and R.vehicleid = V.vehicleid and R.custid = C.custid and R.rentaltype = 7;



```
sqlite> select distinct name, startdate, returndate, totalamount from vehicle V, customer C, rental R, rate A wher
e R.vehicleid = '19VDE1F3XEE414842' and paymentdate is 'NULL' and R.vehicleid = V.vehicleid and R.custid = C.custi
d and R.rentaltype = 7;
Name         StartDate   ReturnDate  TotalAmount
-----------  ----------  ----------  -----------
G. Clarkson  2019-11-01  2019-11-15  1200
J. Brown     2020-01-01  2020-01-29  2400
```

Result: 2 rows returned

Question 11: Return all customers that they never rent a vehicle.
select custid, name from customer where custid NOT IN(select custid from rental);

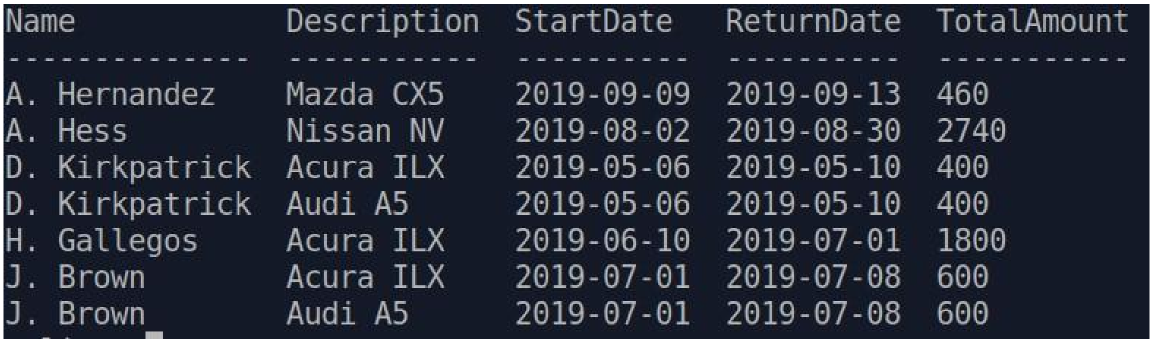


```
CustID  Name
------  ----------------
201     A. Parks
202     S. Patel
204     G. Carver
205     Sh. Byers
206     L. Lutz
207     L. Bernal
208     I. Whyte
209     L. Lott
211     Sh. Dunlap
213     L. Perkins
214     M. Beach
215     C. Pearce
217     M. Lee
218     R. Booker
219     A. Crowther
220     H. Mahoney
222     H. Stokes
223     J. Reeves
224     A. Mcghee
225     L. Mullen
226     R. Armstrong
227     J. Greenaway
228     K. Kaiser Acosta
230     A. Odonnell
231     K. Kay
sqlite> 
```

Result: 25 rows returned

Question 12: Return all rentals that the customer paid on the StartDate. List Customer Name, Vehicle Description, StartDate, ReturnDate, and TotalAmount. Order by Customer Name.

select distinct C.name, V.description, R.startdate, R.returndate, totalamount from rental R, customer C, vehicle V where StartDate = PaymentDate and C.custid = R.custid and V.vehicleid = R.vehicleid ORDER BY Name;



```
Name           Description  StartDate   ReturnDate  TotalAmount
-------------  -----------  ----------  ----------  -----------
A. Hernandez   Mazda CX5    2019-09-09  2019-09-13  460
A. Hess        Nissan NV    2019-08-02  2019-08-30  2740
D. Kirkpatrick Acura ILX    2019-05-06  2019-05-10  400
D. Kirkpatrick Audi A5      2019-05-06  2019-05-10  400
H. Gallegos    Acura ILX    2019-06-10  2019-07-01  1800
J. Brown       Acura ILX    2019-07-01  2019-07-08  600
J. Brown       Audi A5      2019-07-01  2019-07-08  600
```

Result: 7 rows returned

Contributions

Shivam Patel: Task 1, Task 2, Task 3: Queries 1-4

Gerald Kimeu: Task 1, Task 2, Task 3: Queries 9-12

Carlos Uresti: Task 1, Task 2, Task 3: Queries 5-8